

**UNITED STATES DISTRICT COURT  
DISTRICT OF MASSACHUSETTS**

SINGULAR COMPUTING LLC,

Plaintiff,

v.

GOOGLE LLC,

Defendant.

Civil Action No. 1:21-cv-12110

**JURY TRIAL DEMANDED**

**AMENDED COMPLAINT FOR PATENT INFRINGEMENT**

Plaintiff, Singular Computing LLC (“Singular”), for its amended complaint against defendant, Google LLC (“Google”), alleges as follows:

**THE PARTIES**

1. Singular is a Delaware limited liability company having its principal places of business at 10 Regent Street, Newton, Massachusetts 02465 and The Cambridge Innovation Center, 1 Broadway, Cambridge, Massachusetts 02142.

2. Google is a Delaware limited liability company having regular and established places of business in this District, including a major office complex in Cambridge, Massachusetts with over 1,500 employees. Google may be served with process through its registered agent, Corporation Service Company, 84 State Street, Boston, Massachusetts 02109.

**JURISDICTION**

3. This is a civil action for patent infringement under the patent laws of the United States, 35 U.S.C. §§ 271, *et seq.* This Court has subject matter jurisdiction under 28 U.S.C. §§ 1331 and 1338(a).

4. This Court has general personal jurisdiction over Google because Google is engaged in substantial and continuous activity, which is not isolated, at its regular and

established places of business within this judicial district. This Court has specific personal jurisdiction over Google because Google has also committed acts of infringement within this judicial district giving rise to this action and has established more than minimum contacts within this judicial district such that the exercise of jurisdiction over Google by this Court would not offend traditional notions of fair play and substantial justice.

5. Venue is proper in this judicial district pursuant to 28 U.S.C. §§ 1391(b)-(c) and 1400(b) because Google maintains regular and established places of business and has committed acts of patent infringement within this judicial district.

### **FACTUAL BACKGROUND**

6. Singular was founded by Dr. Joseph Bates to, *inter alia*, design, develop, and produce computers having new architectures, including the patented computer architectures at issue in this case. Dr. Bates is the President and Chief Executive Officer of Singular. Since 2009, Singular has continuously operated out of the Boston area.

7. Dr. Bates's interest in computer science dates back to at least 1969, when, at the age of thirteen, he was admitted to Johns Hopkins University as an undergraduate. His success in college sparked a pilot program for exceptionally gifted youths, which led to the widely-recognized Johns Hopkins Center for Talented Youth (also known as "CTY"; *see* <https://cty.jhu.edu> ) that has contributed to the intellectual development of over 165,000 academically advanced pre-college students, including Google founder Sergey Brin. By the age of 17, Dr. Bates had earned bachelor's and master's degrees from Johns Hopkins, both in the field of Computer Science. He continued his studies at Cornell University, where he earned his doctorate in Computer Science when he was 23 years old. Dr. Bates's research and teaching interests have centered around several cutting-edge computer science topics, including formal

logic, the design and implementation of computer programming languages, and artificial intelligence (“AI”).

8. During his career working at the vanguard of computer science, Dr. Bates realized that, although the theoretical computing power inside computers (as represented by the number of transistors inside a computer) was growing exponentially under a phenomenon known as Moore’s Law, the vast majority of that increase in computing power was not being made available to users. With then-existing computer architectures, even computers containing over a billion transistors were designed to typically perform only a handful of operations per unit of time (“clock cycle,” “cycle” or “period”) when using CPUs. Such conventional computers of the time typically performed only a few hundred operations per cycle when using GPUs.

9. In the course of his work, Dr. Bates realized that existing computing architectures prevented computers from achieving their full potential. Computers perform operations using *transistors*, semiconductor devices that control the flow of electric current. There is a correlation between the performance of a computer and the number of transistors contained in the computer. For the last 50 years, due to advances in semiconductor technology, the number of transistors inside computers has generally grown at an exponential rate, doubling roughly every two years, which meant the performance of computers has significantly increased. Computer chips in the early 1970s contained just a few thousand transistors, while many similar chips used today have over 10 *billion* transistors. Dr. Bates recognized, however, that computing power (as measured by the number of operations a computer performs each second, for example) had not increased at the same rate. Dr. Bates further recognized that computing power gains were lagging transistor count gains because a computer built using a conventional architecture, even though it included more transistors, did not use those transistors efficiently.

10. Dr. Bates devised improved computer architectures that allow a computer to make more efficient use of its physical resources (*e.g.*, its transistors). The novel architectures invented by Dr. Bates involve computer chips with contain processing elements purposely designed to perform low precision operations at high dynamic range. By being purposely so designed, numerical values can be represented and manipulated inside processing elements using smaller bit widths (at the cost of lower precision), which in turn enables such processing elements to be smaller than processing elements that perform traditional precision operations (*e.g.*, 32-bit or 64-bit floating point arithmetic). The relatively small size of such processing elements enables a great number of them to be packed inside a computer chip, and operated in parallel with each other, which increases the number of operations performed per cycle by that chip. These architectures thus allow computers to use a given number of transistors more efficiently, while maintaining high dynamic range so as to have broad applicability to a wide variety of computing applications. In particular, Dr. Bates's inventions have revolutionized the field of AI and his patented architectures have vastly increased the speed and performance of computer processors when executing AI applications.

11. A key difference between conventional computer architectures and Dr. Bates's invention relates to a computer's performance of arithmetic operations such as multiplication. In a conventional computer architecture, a typical multiplier circuit inside a processing element contains on the order of a hundred thousand transistors or more. A computer built using Dr. Bates's patented architecture, on the other hand, includes processing elements whose multiplier circuits require a far smaller number of transistors, making it possible to include a very large number of them on a single chip, thereby increasing the number of multiplication operations per cycle the computer is able to perform. Indeed, a computer that uses Dr. Bates's invention can

potentially perform hundreds of times more multiplication operations per cycle, and therefore hundreds of times more multiplication operations per second, than a conventional computer with the same number of transistors.

12. In some embodiments of Singular's patented novel computer architectures, a relatively large number of such processing elements that operate at low precision can be deployed in conjunction with far smaller numbers of relatively larger traditional precision processing elements (*e.g.*, processing elements that represent and manipulate numerical values using an Institute of Electrical and Electronics Engineers (IEEE) standardized "basic format", which formats have a minimum width of 32 bits).

13. Singular's revolutionary approach to computer architecture is described in a provisional patent application entitled "Massively Parallel Processing with Compact Arithmetic Element" that was filed in June of 2009 and made public in June of 2010.

14. After Dr. Bates filed this provisional patent application, he built a prototype computer based on the novel architecture disclosed therein. The Singular prototype was able to execute a software program that, for example, was able to perform neural network image classification thirty times faster than a conventional computer having comparable physical characteristics in terms of its number of transistors, its semiconductor fabrication process and its power draw.

15. As Singular was building prototypes of its new computer, Google belatedly recognized the limitations of its conventional computer architectures in providing users with computer-based services such as Translate, Photos, Search (including Image Search), Assistant, and Gmail. According to Google, these limitations caused a "scary and daunting" situation for Google. The situation arose as Google was starting to deliver these computer-based services by

running AI software programs on its conventional computers. The situation was “scary and daunting” because the new AI software programs required far more computer operations per cycle than the software programs Google was previously executing to deliver such services. For example, by Google’s own estimation, applying its new AI software programs to speech recognition services alone (*e.g.*, Translate and Assistant) would increase the number of operations per cycle required of Google’s computers so drastically that Google would have to at least double its total computing footprint.

16. Google realized it needed to use Dr. Bates’s computer architectures to increase the number of computer operations per cycle executed by its computers. To this end, it copied Dr. Bates’ architecture into the Tensor Processing Units (“TPU”) v2, v3 and v4 devices, also known more generally as Cloud TPU, (together “accused TPUs” or “accused TPU computers”) to deliver, as published by Google, services such as Translate, Photos, Search, Assistant, Cloud and Gmail to the public.

Cloud TPU is the custom-designed machine learning ASIC that powers Google products like Translate, Photos, Search, Assistant, and Gmail. Here’s how you can put the TPU and machine learning to work accelerating your company’s success, especially at scale.

See <https://cloud.google.com/tpu>.

Google drives the public’s use of these services to enhance at least its Ads platform which, in turn, generates at least tens of billions of dollars per year in profit for Google. See <https://www.statista.com/statistics/266206/googles-annual-global-revenue/>.

17. As of 2017, Google housed its accused TPU computers in the United States in at least eight data centers. As of 2017, the approximate cost to build each data center was at least \$1.5 billion. As Google recognized, unless it incorporated Dr. Bates’s patented technology, it

would have had to at least double its number of data centers in the U.S. to sixteen. Assuming a cost of \$1.5 billion per new data center, this would have cost Google a total of at least \$12 billion.

18. With the steep growth of its business since 2017, Google now maintains at least fourteen data centers in the United States for its accused TPU computers. *See* [www.google.com/about/datacenters/locations/](http://www.google.com/about/datacenters/locations/). The accused TPUs are installed and operated by Google in one or more of Google's data centers located at: Berkeley County, South Carolina; Council Bluffs, Iowa; The Dalles, Oregon; Douglas County, Georgia; Henderson, Nevada; Jackson County, Alabama; Lenior, North Carolina; Loudoun County, Virginia; Mayes County, Oklahoma; Midlothian, Texas; Montgomery County, Tennessee; New Albany, Ohio; Papillon, Nebraska, and Storey County, Nevada.

19. In the Securities and Exchange Commission Form 10-K filed by Google's parent Alphabet, Inc. ("Alphabet") for the fiscal year ending December 31, 2020, Alphabet reported net income of approximately \$40.2 billion on revenues exceeding \$182 billion.

### **THE PATENTS-IN-SUIT**

20. On August 25, 2020, the United States Patent and Trademark Office ("USPTO") issued United States Patent No. 10,754,616, titled PROCESSING WITH COMPACT ARITHMETIC PROCESSING ELEMENT ("the '616 patent"). The '616 patent is valid and enforceable.

21. On November 9, 2021, the USPTO issued United States Patent No. 11,169,775, titled PROCESSING WITH COMPACT ARITHMETIC PROCESSING ELEMENT ("the '775 patent"). The '775 patent is valid and enforceable.

22. The application to which the '616 patent and the '775 patent claim priority (No. 61/218,691) was filed on June 19, 2009.

23. Singular is the owner and assignee of all rights, title and interest in and to the '616 patent and the '775 patent, and holds all substantial rights therein, including the rights to grant licenses, to exclude others, and to enforce and recover past damages for infringement.

24. The claims asserted in this action are eligible for patenting under 35 U.S.C. § 101.

25. Claim 10 of the '616 patent recites the following limitations, each of which is found in the accused TPUs as set forth below:<sup>1</sup>

*10. A computing system, comprising:*

*a host computer;*

*a computing chip comprising:*

*a processing element array comprising a plurality of first processing elements, wherein the plurality of first processing elements is no less than 5000 in number, wherein each of a first subset of the plurality of first processing elements is positioned at a first edge of the processing element array, and wherein each of a second subset of the plurality of first processing elements is positioned in the interior of the processing element array;*

*an input-output unit connected to each of the first subset of the plurality of first processing elements;*

*a plurality of processing element connections, each processing element connection connecting one of the plurality of first processing elements with another of the plurality of first processing elements, wherein each of the plurality of first processing elements is connected to at least one other of the plurality of first processing elements by at least one of the plurality of processing element connections;*

*a plurality of memory units, wherein each of the plurality of first processing elements is associated with a corresponding one of the plurality of memory units, and wherein each of the plurality of memory units is local to its associated one of the plurality of first processing elements; and,*

---

<sup>1</sup> Claim 10 of the '616 patent is a dependent claim; it depends from claim 8, which in turn depends from independent claim 7. It has been written herein in independent form, to include the limitations of claims 7 and 8 from which it depends.



*a plurality of arithmetic units, wherein each of the plurality of first processing elements has positioned therein at least one of the plurality of arithmetic units; and,*

*a host connection at least partially connecting the input-output unit with the host computer;*

*wherein the plurality of arithmetic units each comprises a first corresponding multiplier circuit adapted to receive as a first input to the first corresponding multiplier circuit a first floating point value having a first binary mantissa of width no more than 11 bits and a first binary exponent of width at least 6 bits, and to receive as a second input to the first corresponding multiplier circuit a second floating point value having a second binary mantissa of width no more than 11 bits and a second binary exponent of width at least 6 bits;*

*wherein the computing chip further comprises a plurality of second processing elements, wherein the plurality of second processing elements each comprises a second corresponding multiplier circuit adapted to receive as inputs to the second corresponding multiplier circuit two floating point values each of width at least 32 bits;*

*wherein, other than the plurality of second processing elements, the computing chip has no other processing element that comprises a multiplier circuit adapted to receive as inputs to the multiplier circuit two floating point values each of width at least 32 bits;*

*wherein the plurality of first processing elements is greater in number, by at least 100, than the plurality of second processing elements; and*

*wherein said host computer is programmed to provide instructions to said computing chip that, when executed, cause said processing element array to perform an operation whose output is used to identify at least one image, from a plurality of images to be searched, that is similar to at least one input image.*

26. Claim 1 of the '775 patent recites the following limitations, each of which is likewise found in the accused TPUs as set forth below:

*1. A computing system, comprising:*

*a host computer;*

*a computing chip comprising:*

*a processing element array comprising a first edge processing element positioned at a first edge of the processing element array, a second edge processing element positioned at the first edge of the processing element array, a first interior processing element positioned at a first location in the interior of the processing element array, and a second interior processing element positioned at a second location in the interior of the processing element array;*

*a first processing element connection connecting the first edge processing element with the first interior processing element;*

*a second processing element connection connecting the second edge processing element with the second interior processing element;*  
*an input-output unit connected to the first edge processing element and the second edge processing element;*  
*a first memory local to the first edge processing element;*  
*a second memory local to the second edge processing element;*  
*a third memory local to the first interior processing element;*  
*a fourth memory local to the second interior processing element; and,*  
*a fifth arithmetic unit;*  
*wherein the first edge processing element comprises a first arithmetic unit;*  
*wherein the second edge processing element comprises a second arithmetic unit;*  
*wherein the first interior processing element comprises a third arithmetic unit; and*  
*wherein the second interior processing element comprises a fourth arithmetic unit; and,*  
*a host connection at least partially connecting the input-output unit with the host computer;*  
*wherein the first, second, third and fourth arithmetic units each comprises a corresponding multiplier circuit adapted to receive as a first input to the corresponding multiplier circuit a first floating point value having a first binary mantissa of width no more than 11 bits and a first binary exponent of width at least 6 bits, and to receive as a second input to the corresponding multiplier circuit a second floating point value having a second binary mantissa of width no more than 11 bits and a second binary exponent of width at least 6 bits;*  
*wherein the fifth arithmetic unit comprises a corresponding multiplier circuit adapted to receive as inputs to the corresponding multiplier circuit two floating point values each of width at least 32 bits;*  
*wherein the multiplier circuit corresponding to the first arithmetic unit comprises a first plurality of transistors and has no other transistors, the multiplier circuit corresponding to the second arithmetic unit comprises a second plurality of transistors and has no other transistors, the multiplier circuit corresponding to the third arithmetic unit comprises a third plurality of transistors and has no other transistors, the multiplier circuit corresponding to the fourth arithmetic unit comprises a fourth plurality of transistors and has no other transistors, and the multiplier circuit corresponding to the fifth arithmetic unit comprises a fifth plurality of transistors; and,*  
*wherein the fifth plurality of transistors exceeds in number each of the first plurality of transistors, the second plurality of transistors, the third plurality of transistors, and the fourth plurality of transistors.*

27. The inventions recited in claim 10 of the '616 patent and claim 1 of the '775 patent (together “the Asserted Claims”) were not conventional. Actually reducing the claimed inventions to practice required the design and manufacture of a computer that was fundamentally different from prior art computers. Existing prior art computers did not practice the invention, nor could they be easily reconfigured or modified to do so, because computer hardware of the time was unsuitable for implementing Dr. Bates’s inventions.

28. Computers built using the novel architecture of the Asserted Claims have many advantages over computers built using conventional architectures. These advantages include, but are not limited to, the following combination of features:

- a) the inclusion of many more processing elements with multiplier circuits on a single computer chip having a given set of resources, such as transistors, than prior art computer chips having a similar set of resources, by utilizing relatively imprecise multiplication circuits that represent and manipulate high dynamic numerical values using smaller mantissa bit widths and thus require far fewer transistors than conventional, traditional-precision multiplication circuits;
- b) the performance of a far greater number of operations per cycle—potentially on the order of 100 times or more—than a conventional computer of the time having the same number of transistors, semiconductor fabrication process and power draw; and
- c) the support of software programs that require operations to be performed on numbers having high dynamic range.

29. Computers built using the novel architecture of the Asserted Claims have (i) a relatively large number of smaller lower-precision processing elements that each represent numerical values using smaller mantissa bit widths, and (ii) a relatively smaller number of larger

traditional-precision processing elements. For example, the claims recite a number of processing elements each comprising a multiplier circuit that is adapted to receive as inputs floating point values having a binary mantissa no more than 11 bits wide, and a far smaller number of processing elements each comprising a multiplier circuit that is adapted to receive as inputs traditional-precision floating point values.

30. Collectively, the inventions recited in the Asserted Claims provide many advantages over the prior art. For example, the claimed systems use transistors more efficiently than those of the prior art, which allows them to perform on the order of 100 times or more operations per cycle than a comparable prior art computer having the same number of transistors.

31. The Asserted Claims also address, *inter alia*, the inefficient use of transistors in prior art computer architectures described above. For example, as stated above, Dr. Bates's patented processing elements including multiplier circuits, each of which utilizes a smaller number of transistors than a traditional-precision multiplier circuit of prior art computer architectures. This difference in the required number of transistors per processing element, which is explicitly recited in the claims, makes it possible to include more of the claimed low-precision processing elements in a computer, which in turn allows the computer to perform many more operations per cycle than a conventional computer having comparable computing resources (*e.g.*, number of transistors, power draw, etc.).

32. Dr. Bates's inventions solve the aforementioned problem of inefficient transistor usage with an unconventional and novel approach to computer architecture that is fundamentally different from prior art computer architectures. Dr. Bates's inventions were not obvious to one of ordinary skill in the art at the time of their invention. Prior art computer architectures did not comprise an overwhelming majority of relatively smaller processing elements that operate at low

precision with high dynamic range, as compared to larger processing elements that operate at traditional precision with high dynamic range. Before Dr. Bates invented it, such a computer was neither previously in existence, nor was it described in any patent or printed publication.

33. Indeed, when the priority application was filed in 2009, the novel architecture invented by Dr. Bates went against a general consensus among those of skill in the art that a computer with a large number of low-precision processing elements was incapable of acceptable performance. It was not obvious, and was in fact counterintuitive, to those skilled in the art as of 2009 to make a computer from a very large number of processing elements that operate at low precision and with high dynamic range, knowing for example that such a computer was going to be used by software programs to execute numerous tasks that each required hundreds, thousands or even millions of sequential arithmetic operations, with each such operation potentially producing errors that could accumulate over time as tasks were executed. Dr. Bates nonetheless conceived of, made, and patented a working computer utilizing such low-precision processing elements, and demonstrated that such a computer could perform better than prior art computers across a variety of applications.

34. The Asserted Claims recite a concrete structure for achieving more efficient computer functionality and are not directed to every way of achieving those results. The architecture described by the Asserted Claims departs from earlier approaches to computer architecture. The Asserted Claims are directed to specific structural features (*e.g.*, imprecise processing elements that can represent and manipulate numerical values using smaller mantissa bit widths) that cause improvements in the capabilities of computing devices (*e.g.*, using the computer's transistors more efficiently, by packing more processing elements into a device's

computer chip, thereby allowing software programs to perform more operations per cycle on that chip).

35. In short, Dr. Bates's fundamentally new, unconventional and novel approach to computer architecture was not obvious, conventional or routine to one of ordinary skill in the art at the time of the invention. In conventional architectures for example, the overwhelming majority of the processing elements were not low-precision processing elements that operated at high-dynamic range; as a matter of fact, prior to Dr. Bates's invention thereof, such a computer did not exist.

36. Computer architects, as of 2009, taught away from Dr. Bates's inventions. Because his processing elements represent and manipulate numerical values using reduced mantissa bit widths, Dr. Bates's claimed processing elements each frequently generate, in response to a request to perform arithmetic operations on high dynamic range numbers, results that materially differ from the exact, accurate results of those operations. In 2009, it was counterintuitive to those of skill in the art to design a computer having processing elements that produce such intentionally imprecise results in executing millions of operations per second, wherein each such operation potentially produces errors that collectively can accumulate over time. Nonetheless, Dr. Bates conceived of and built a working computer that embodies the claimed invention and included a large number of low-precision processing elements.

37. The inventions claimed in the asserted patents ushered in a revolutionary increase in computer efficiency through improved architecture. The Asserted Claims recite architectural elements of computer design such as a relatively larger number of lower precision processing elements and a smaller number of traditional-precision processing elements, that all support high dynamic range.

38. The inventions recited in the Asserted Claims were not conventional or routine as they provided more efficient use of a computer's transistors to perform an increased number of operations per cycle, albeit at reduced precision, while supporting software programs that require operations to be performed on numbers having high dynamic range. Conventional computers, for example, even when designed for execution of AI software programs, did not have such features.

### **COUNT I** **INFRINGEMENT OF THE '775 PATENT**

39. Paragraphs 1-38 above are incorporated herein by reference.

40. As set forth below, Google has directly infringed, and continues to directly infringe, literally and/or by the doctrine of equivalents, at least claim 1 of the '775 patent by making, testing, using, offering for sale, selling and/or importing into the United States the accused TPUs that are used inside Google's existing data centers.

41. The accused TPUs power at least Google Translate, Photos, Search, Assistant and/or Gmail. For example, according to Google:

### **Empowering businesses with Google Cloud AI**

Machine learning has produced business and research breakthroughs ranging from network security to medical diagnoses. We built the Tensor Processing Unit (TPU) in order to make it possible for anyone to achieve similar breakthroughs. Cloud TPU is the custom-designed machine learning ASIC that powers Google products like Translate, Photos, Search, Assistant, and Gmail. Here's how you can put the TPU and machine learning to work accelerating your company's success, especially at scale.

42. According to Google's Chief Executive Officer ("CEO"), Sundar Pichai, Google's accused TPUs have played a "big part" in Google's advances in AI services, are used

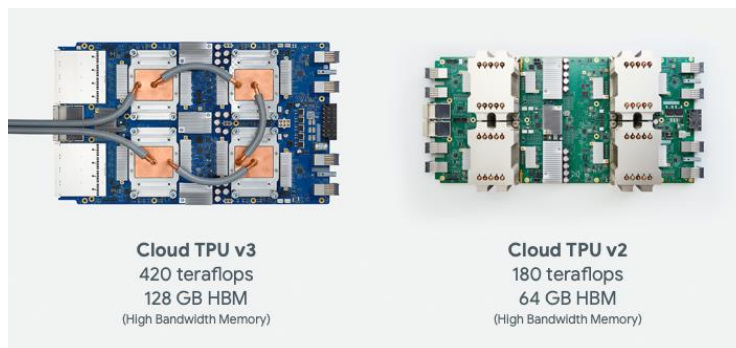
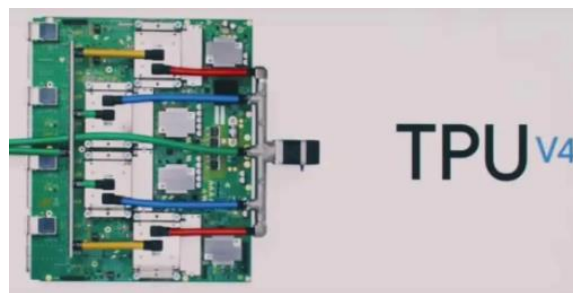


“across all [Google’s] products,” and are used “every time” a Google search is made. *See, e.g.*, [blog.google/technology/developers/io21-helpful-google/](https://blog.google/technology/developers/io21-helpful-google/).

43. According to Google:

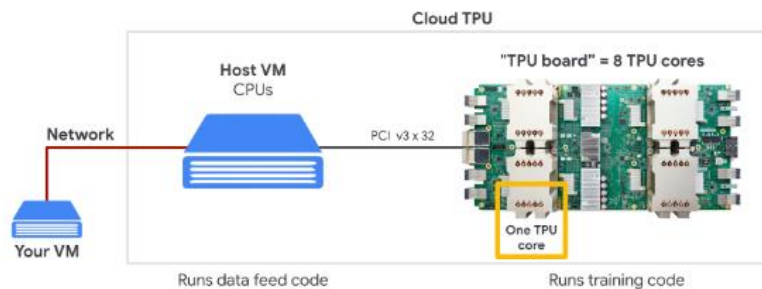
#### TPU versions

Each TPU version defines the specific hardware characteristics of a TPU device. The TPU version defines the architecture for each TPU core, the amount of high-bandwidth memory (HBM) for each TPU core, the interconnects between the cores on each TPU device, and the networking interfaces available for inter-device communication.



#### Cloud TPU

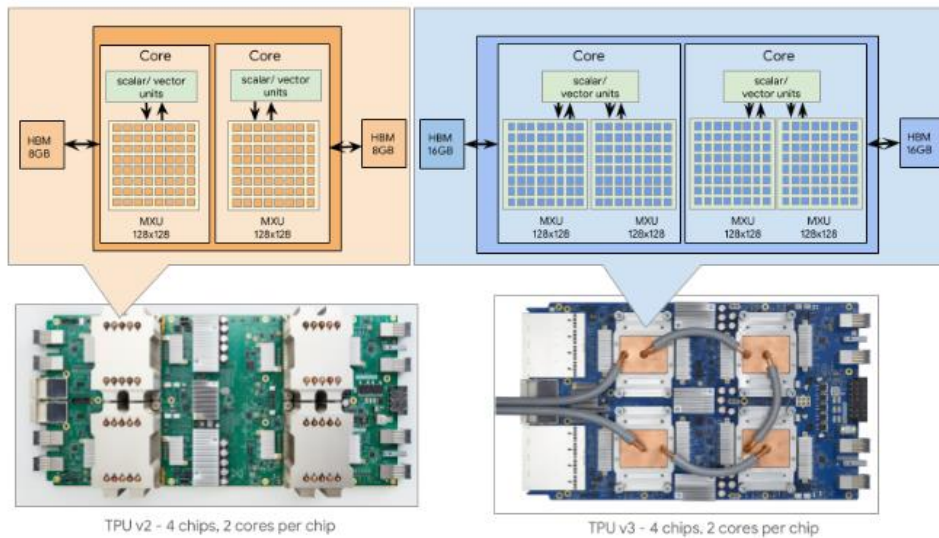
When you request one “[Cloud TPU v2](#)” on Google Cloud Platform, you get a virtual machine (VM) which has a PCI-attached TPU board. The TPU board has four dual-core TPU chips. Each TPU core features a VPU (Vector Processing Unit) and a 128x128 MXU (MatriX multiply Unit). This “Cloud TPU” is then usually connected through the network to the VM that requested it. So the full picture looks like this:





44. Google describes the accused v2 and v3 TPUs, *inter alia*, as follows:

- TPU v2:
  - 8 GiB of HBM for each TPU core
  - One MXU for each TPU core
  - Up to 512 total TPU cores and 4 TiB of total memory in a TPU Pod
- TPU v3:
  - 16 GiB of HBM for each TPU core
  - Two MXUs for each TPU core
  - Up to 2048 total TPU cores and 32 TiB of total memory in a TPU Pod

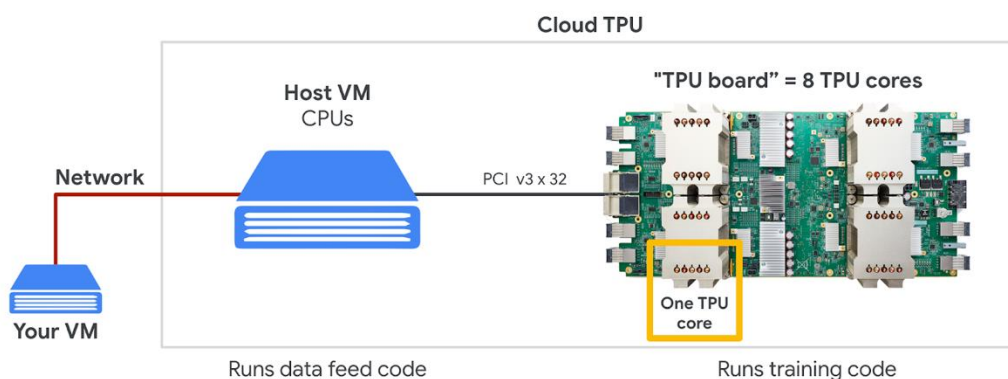


45. According to cloud.google.com, “TPU v4 Pods are already widely deployed throughout Google data centers for [Google’s] internal machine learning workloads and will be available via Google Cloud later this year.” *See also* the following from <https://jonathan-hui.medium.com/ai-chips-tpu-3fa0b2451a2d>:

*Google's fourth-generation TPU ASIC offers more than double the matrix multiplication TFLOPs of TPU v3, a significant boost in memory bandwidth, and advances in interconnect technology. Google's TPU v4 MLPerf submissions take advantage of these new hardware features with complementary compiler and modeling advances. The results demonstrate an average improvement of 2.7 times over TPU v3 performance at a similar scale in the last MLPerf Training competition.*

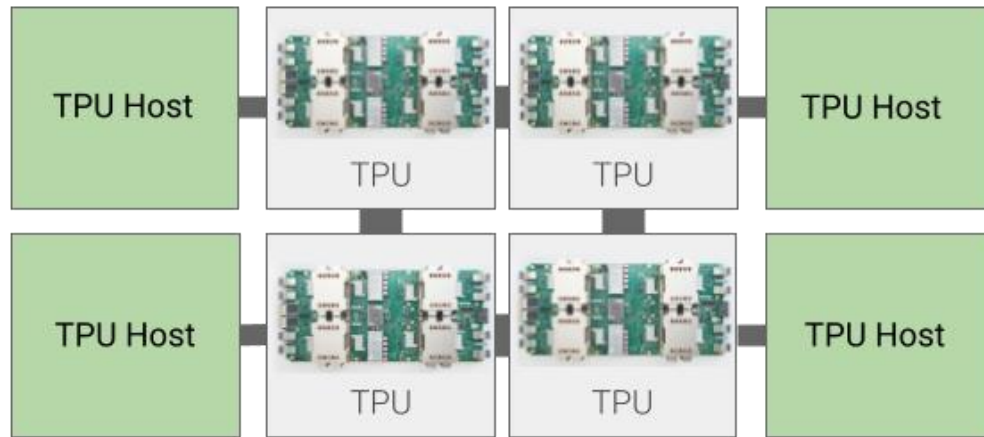
TPU v4 double the matrix multiplication TFLOPs and offers a significant boost in memory bandwidth from new interconnect technology.

46. As published by Google, each of the accused TPUs is a computer system comprising one host computer, namely a Host VM CPU ("TPU host"), and at least one TPU board. The TPU host is connected to each TPU board, and each TPU board in turn comprises one or more TPU computing chips ("TPU chips"). Each TPU chip in turn comprises a plurality of TPU cores. Each TPU board is connected to the TPU host for loading and preprocessing data for feeding into the TPU cores. *See* <https://codelabs.developers.google.com/codelabs/keras-flowers-data#2>:



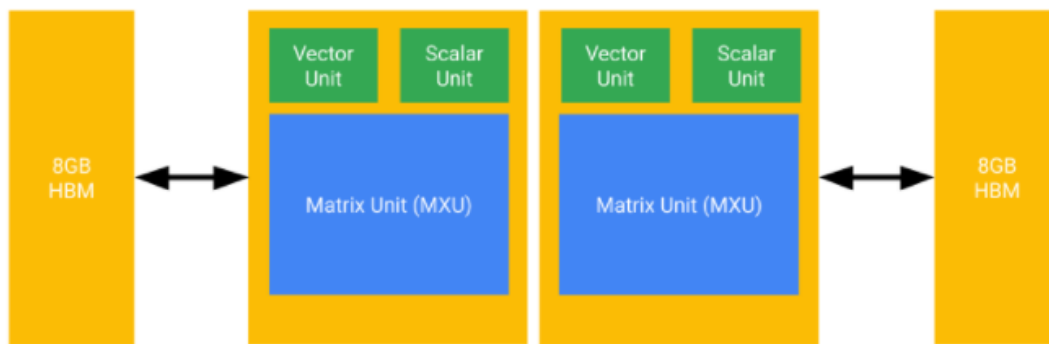
47. As published by Google, and shown a few paragraphs above, a TPU v3 Pod, for example, may have up to 2,048 TPU cores and 32 TiB of memory, as shown a few paragraphs above. According to Google's CEO, Sundar Pichai, a TPU v4 Pod has 4,096 v4 chips and is

capable of executing one quintillion floating-point operations per second (*see* remarks delivered by Mr. Pichai at the Google I/O 2021 conference).

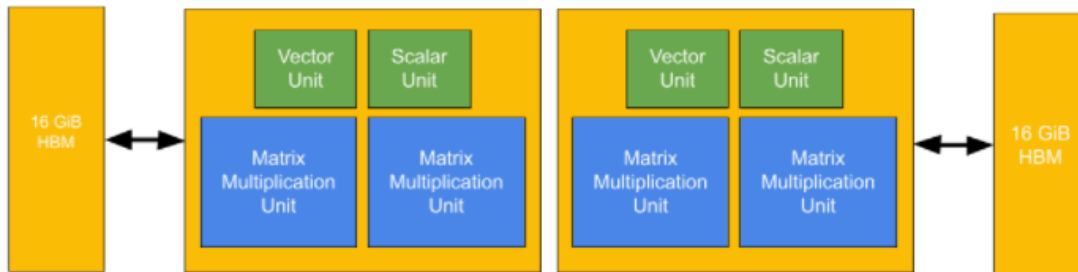


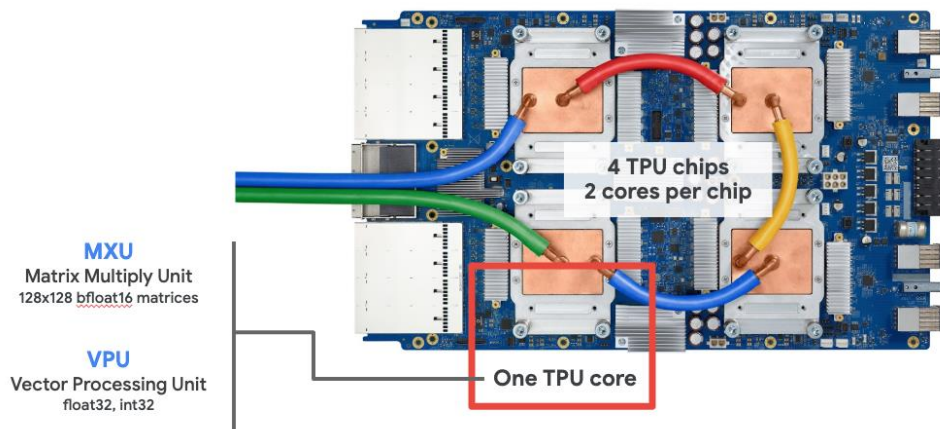
48. As shown by Google’s own publications, each TPU core comprises a Matrix Multiply Unit array (“MXU array”) that performs matrix multiplication operations, a Vector Unit (also known as a Vector Processing Unit, or “VPU”) and a Scalar Unit.

TPU v2:

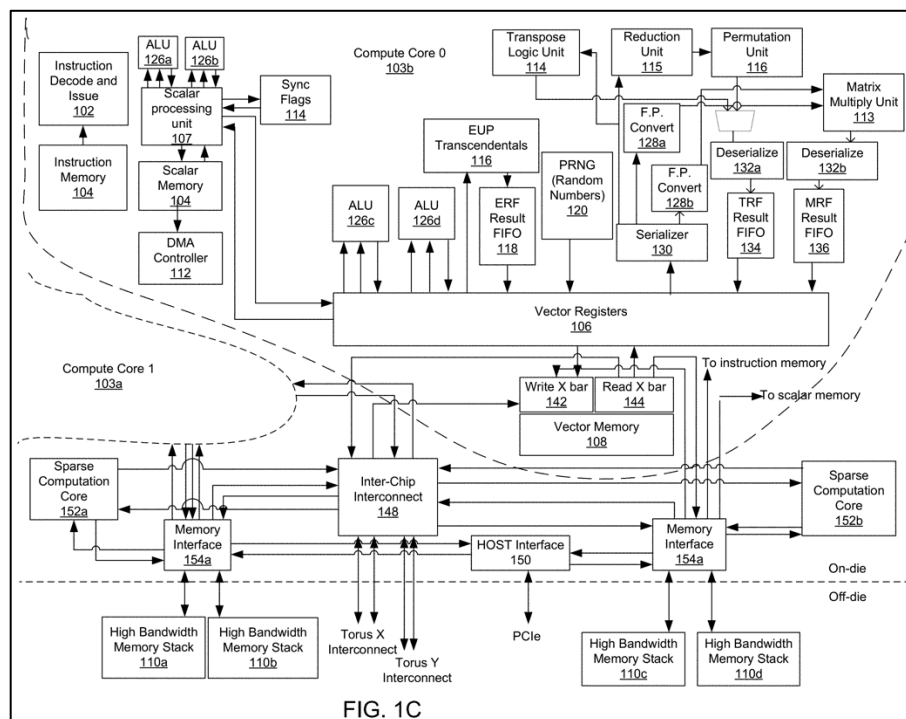
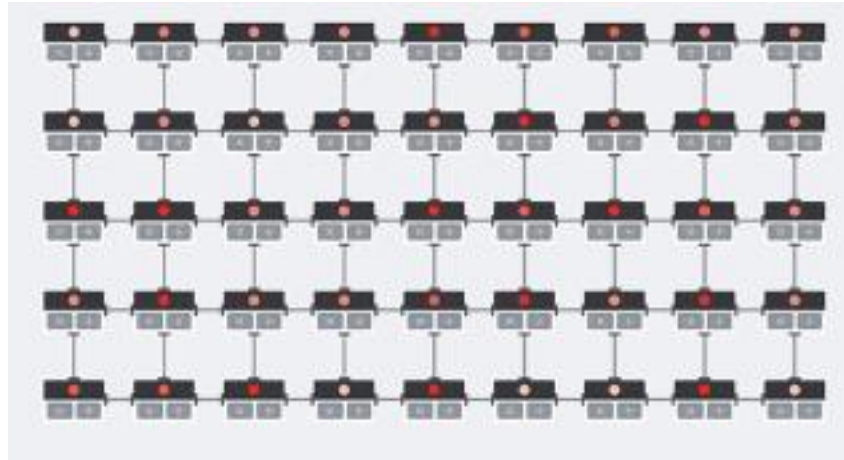


TPU v3:

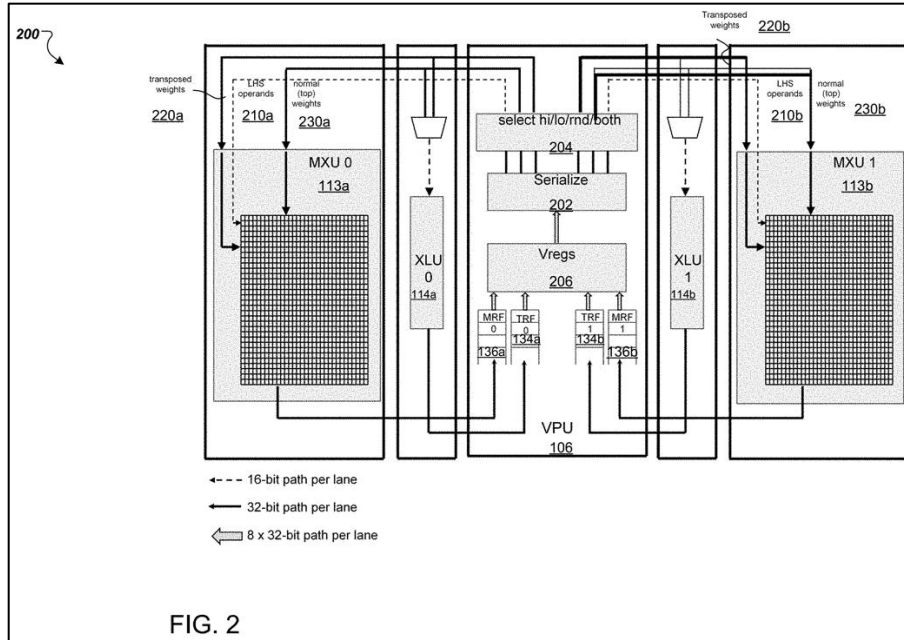




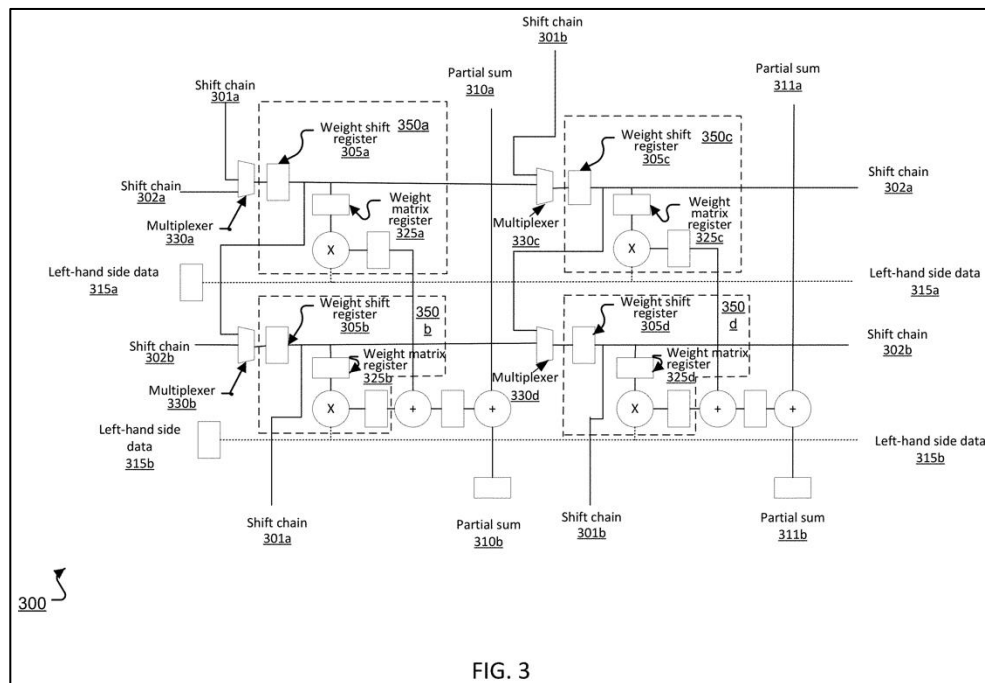
49. As published by Google, each of the aforementioned TPU chips, comprises a processing element array that includes at least two processing elements each positioned as part of a column at the left edge of the array, and a plurality of at least two connected adjacent processing elements each positioned at the interior of the aforementioned processing element array and to the right of the at least two aforementioned left edge processing elements. As also shown, a processing element connection connects each said left edge processing element to a respective one of said interior processing elements. Figure 1C, Figure 2 and Figure 3 below are taken from Google U.S. Patent No. 10,621,269 (“Google ’269 patent”), whose specification has been represented by Google as being reflective of the architecture of the accused TPUs’ chips. As represented in the Google ’269 patent, Figure 3 illustrates a “multi-cell inside a matrix multiply unit.” *See also Introduction to Cloud TPU* (<https://cloud.google.com/tpu/docs/intro-to-tpu>).



Google '269 Patent, Fig. 1C (showing a “neural network processing system”)



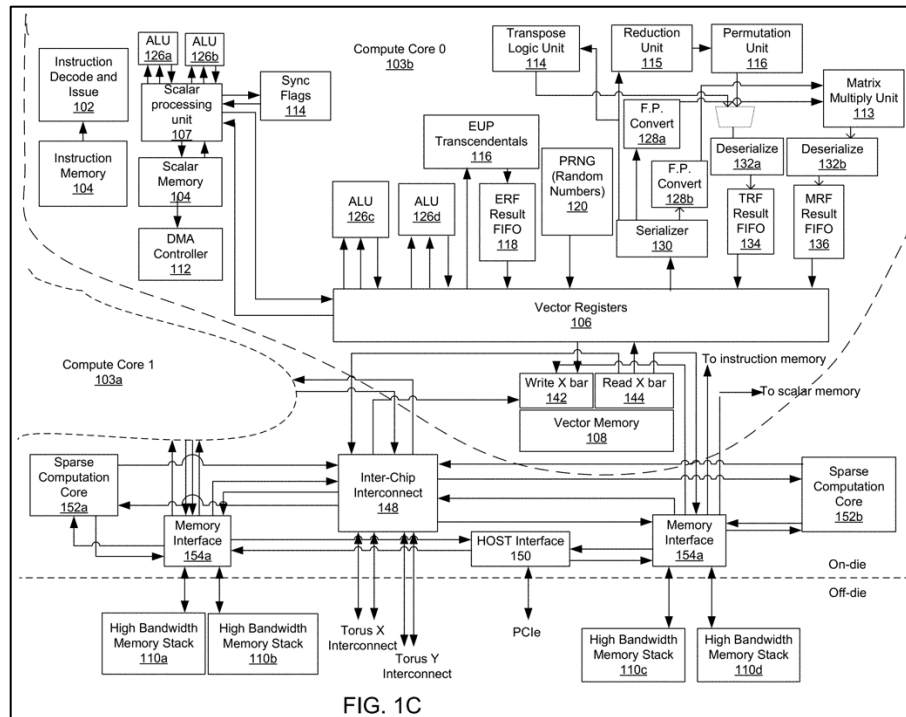
Google '269 Patent, Fig. 2 (showing a “two-dimensional systolic array”)



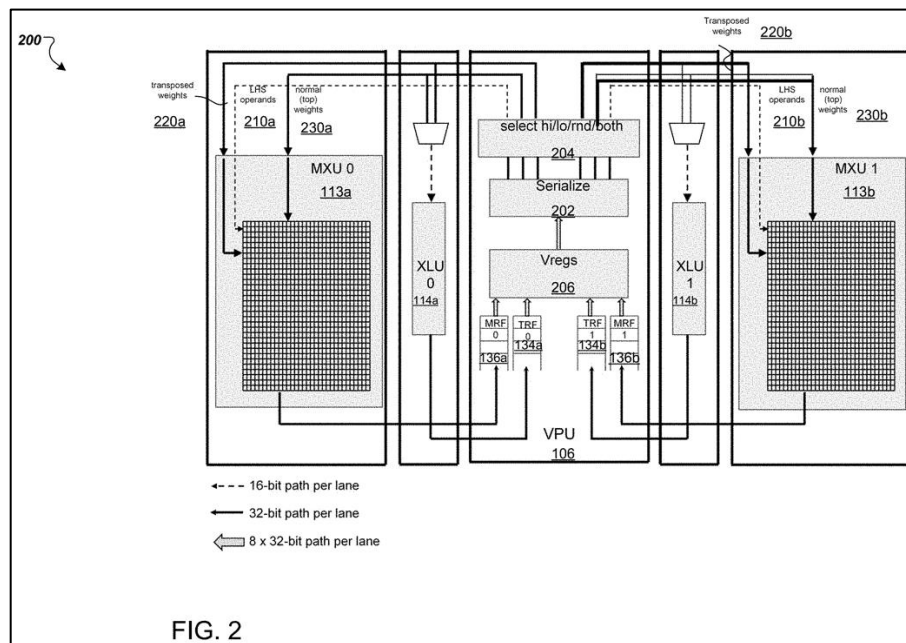
Google '269 Patent, Fig. 3 (showing a “multi-cell inside a systolic array”)

50. As illustrated in Figure 1C and Figure 2 of the Google '269 patent, each accused TPU comprises a TPU chip that itself comprises at least one input-output unit (“TPU input-output unit”) that is connected to the aforementioned left edge processing elements:



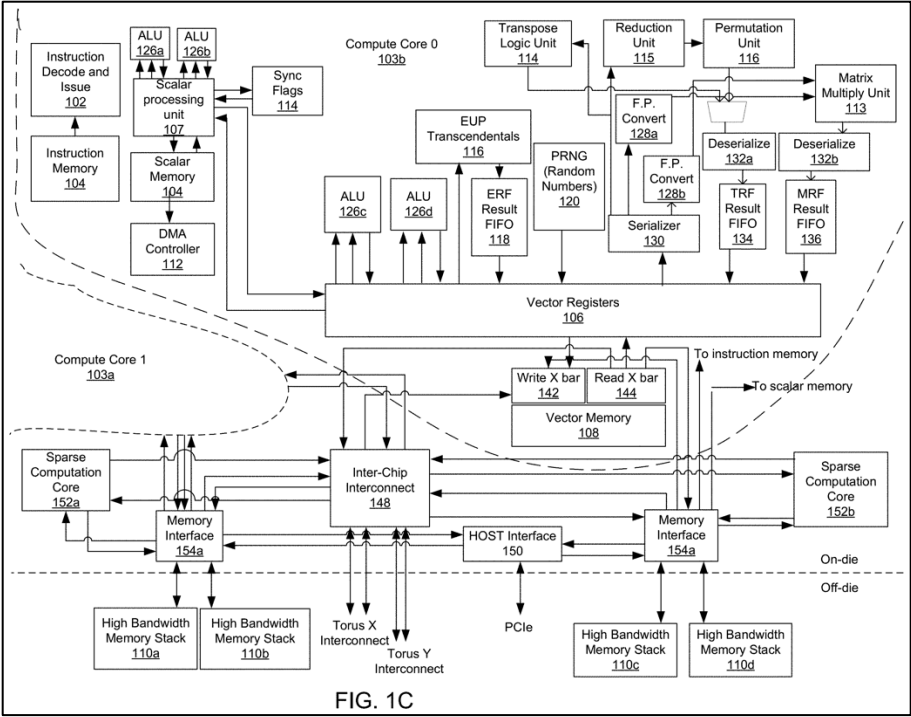


Google '269 Patent, Fig. 1C (showing a “neural network processing system”)

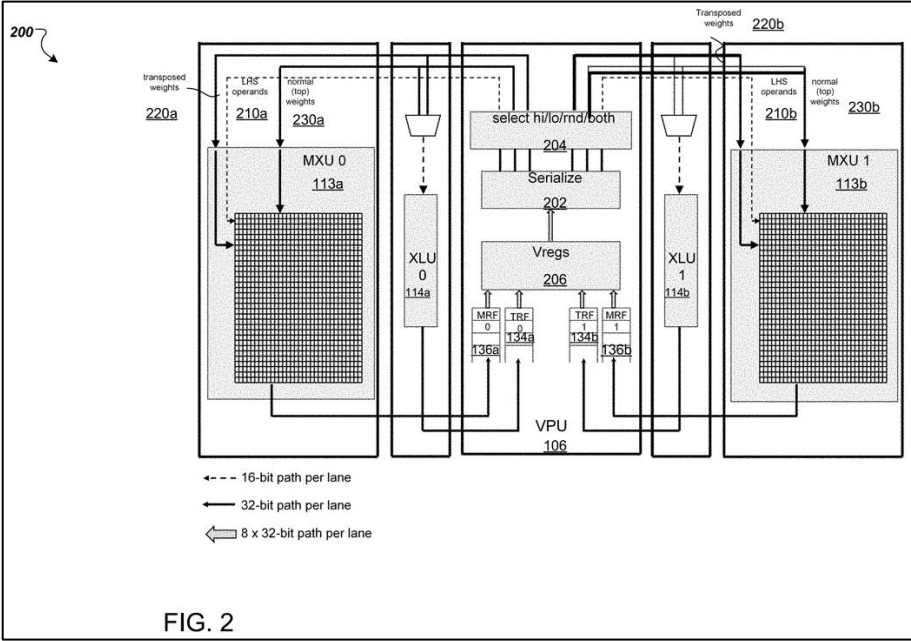


Google '269 Patent, Fig. 2 (showing a “two-dimensional systolic array”)

51. As shown in Figure 3 of the Google '269 patent, each accused TPU comprises a TPU chip that itself comprises a plurality of memories that are each local to one of the aforementioned processing elements.



Google '269 Patent, Fig. 1C (showing a “neural network processing system”)



Google '269 Patent, Fig. 2 (showing a “two-dimensional systolic array”)



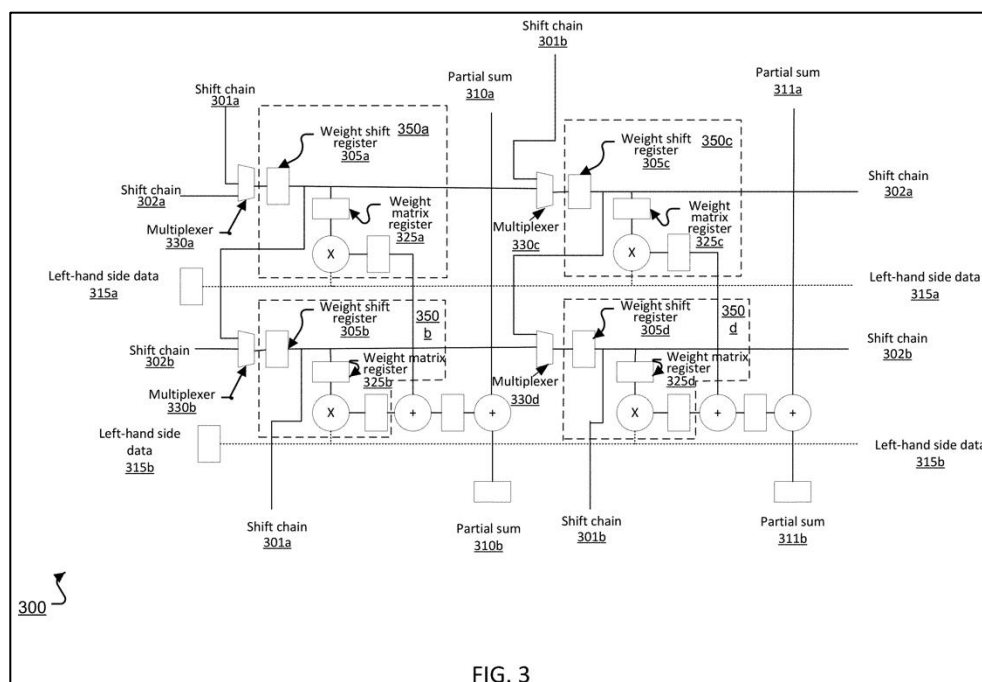


FIG. 3

Google '269 Patent, Fig. 3 (showing a “multi-cell inside a systolic array”)

52. As shown in Figure 2 and Figure 3 of the Google '269 patent, each of the aforementioned processing elements comprises one arithmetic unit (“MXU arithmetic unit”), each of which in turn comprises one multiplier circuit (“MXU multiplier circuit”).

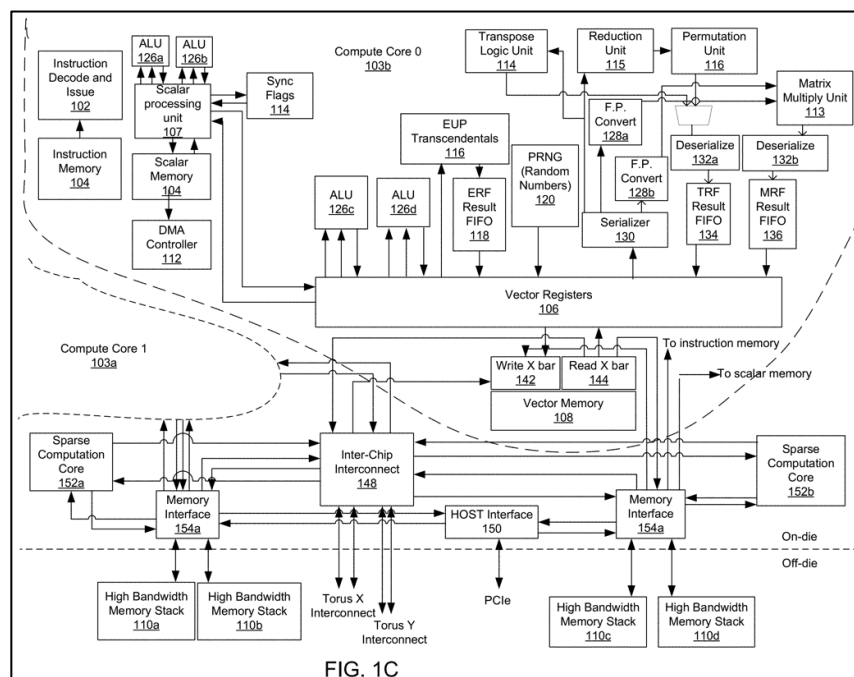
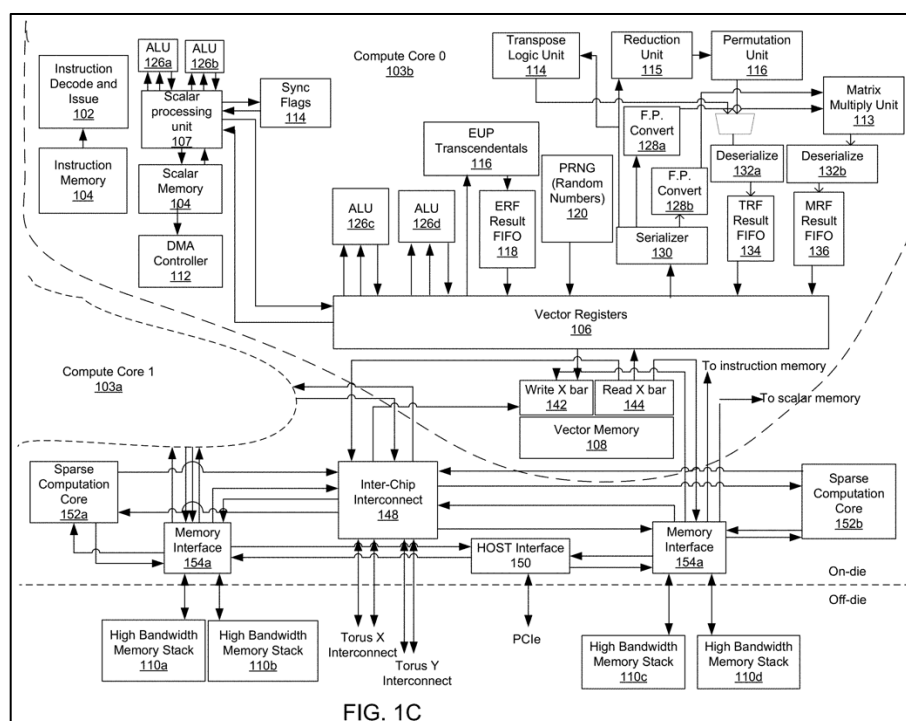
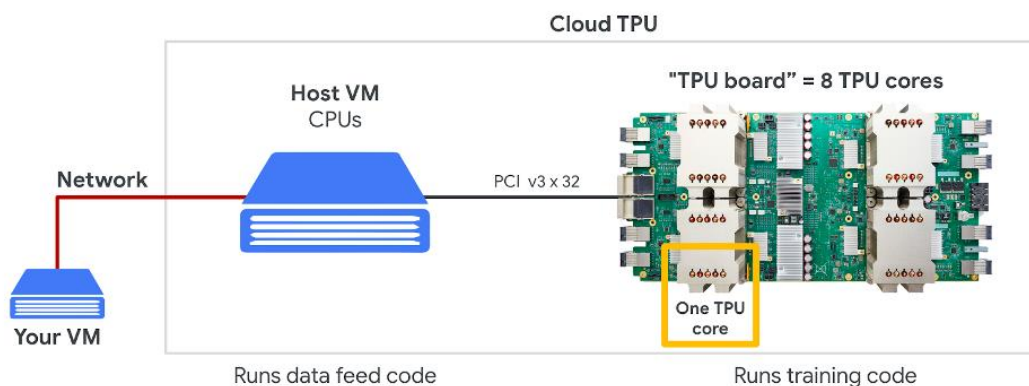


FIG. 1C

Google '269 Patent, Fig. 1C (showing a “neural network processing system”)



<https://cloud.google.com/tpu/docs/>. The aforementioned host connection at least partially connects the aforementioned TPU input-output unit with the aforementioned TPU host.



Google '269 Patent, Fig. 1C (showing a “neural network processing system”)

54. Each of the aforementioned MXU multiplier circuits inside each of the aforementioned MXU arithmetic units is adapted to receive as inputs two floating point values having a “bfloat16” format, as described in the following paragraphs, excerpted from Google’s own published documentation:

Cloud TPU

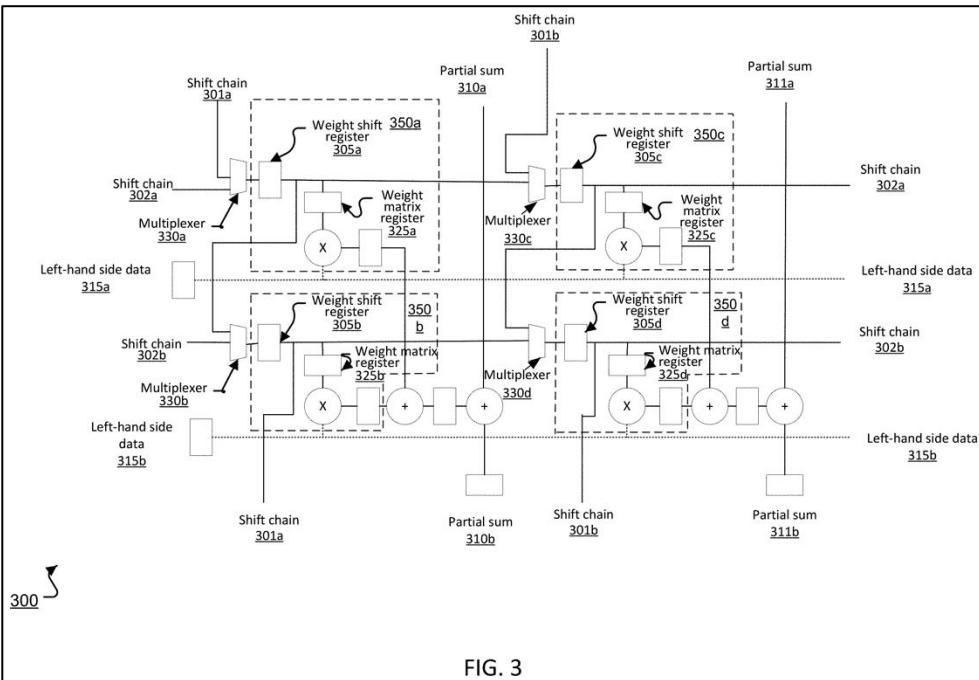
## System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced **bfloat16** precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE **half-precision** representation.

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

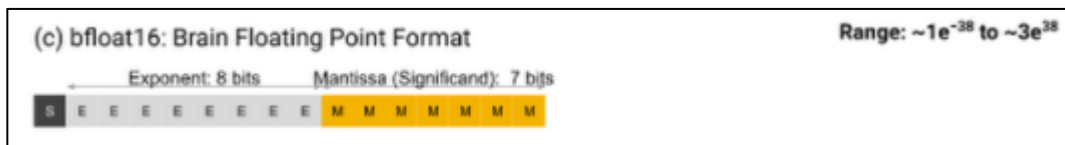
## Choosing bfloat16

Our hardware teams chose bfloat16 for Cloud TPUs to improve hardware efficiency while maintaining the ability to train accurate deep learning models, all with minimal switching costs from FP32. The physical size of a hardware multiplier scales with the *square* of the mantissa width. With fewer mantissa bits than FP16, the bfloat16 multipliers are about half the size in silicon of a typical FP16 multiplier, and they are *eight times* smaller than an FP32 multiplier!



Google '269 Patent, Fig. 3 (showing a “multi-cell inside a systolic array”)

55. According to Google’s published documents, the bfloat16 format characterizing the input received by each MXU multiplier circuit of the accused TPUs, has a sign bit, 8 exponent bits and 7 mantissa bits:



See <https://cloud.google.com/tpu/docs/bfloat16>.

56. As Google explains in the material cited above, the bfloat16 format utilizes a binary mantissa of width that is no more than 11 bits and a binary exponent of width that is at least 6 bits. Google copied the idea of a computer device having an array of processing elements that perform floating-point arithmetic using such a number format, from Dr. Bates.

57. Each of the aforementioned VPUs in the aforementioned TPU chips, comprises at least one arithmetic unit (“VPU arithmetic unit”). Each VPU arithmetic unit comprises at least one multiplier circuit (“VPU multiplier circuit”) that is adapted to receive as inputs two floating point values each of a width that is at least 32 bits wide. *See, e.g.*, <https://codelabs.developers.google.com/codelabs/keras-flowers-data/#2> (“The VPU handles float32 and int32 computations”).

30 The chip stores data in high bandwidth memory (156c-d), reads the data in and out of vector memory (108), and processes the data. The compute core (103b) itself includes a vector memory (108) that is on-chip S-RAM which is divided into two dimensions. The vector memory has  
 35 address space in which addresses hold floating point numbers, i.e., 128 numbers that are each 32-bits. The compute core (103b) also includes a computational unit that computes values and a scalar unit that controls the computational unit.

The vector processor consists of a 2-dimensional array of  
 40 vector processing units, i.e., 128x8, which all execute the same instruction in a single instruction, multiple-data (SIMD) manner. The vector processor has lanes and sub-lanes, i.e., 128 lanes and 8 sublanes. Within the lane, the vector units communicate with each other through load and  
 45 store instructions. Each vector unit can access one 4-byte value at a time. Vector units that do not belong to the same lane cannot communicate directly. These vector units must use the reduction/permutation unit which is described below.

The computational unit includes vector registers, i.e., 32  
 50 vector registers, in a vector processing unit (106) that can be used for both floating point operations and integer operations. The computational unit includes two arithmetic logic units (ALUs) (126c-d) to perform computations. One ALU (126c) performs floating point addition and the other ALU  
 55 (126d) performs floating point multiplication. Both ALUs

Google '269 Patent, 6:30-55

58. Each aforementioned MXU multiplier circuit comprises a first number of transistors, the aforementioned VPU multiplier circuit comprises a second number of transistors, and the first number is less than the second number. Google engineer Jeffrey Dean, the head of Google Brain, expressly admitted this:

Furthermore, one major area & power cost of multiplier circuits for a floating point format with M mantissa bits is the  $(M+1) \times (M+1)$  array of full adders (that are needed for multiplying together the mantissa portions of the two input numbers. The IEEE fp32, IEEE fp16 and bfloat16 formats need 576 full adders, 121 full adders, and 64 full adders, respectively. **Because multipliers for the bfloat16 format require so much less circuitry**, it is possible to put more multipliers in the same chip area and power budget, thereby meaning that ML accelerators employing this format can have higher flops/sec and flops/Watt, all other things being equal.



Dean, Jeffrey. (2020). 1.1 *The Deep Learning Revolution and Its Implications for Computer Architecture and Chip Design*. 8-14. 10.1109/ISSCC19947.2020.9063049 (emphasis added).

This fact was further confirmed in a paper published by the team of Google engineers responsible for designing and building the accused TPUs (including, *inter alia*, Norman Jouppi and David Patterson):

Operation		Picojoules per Operation		
		45 nm	7 nm	45 / 7
+	Int 8	0.03	0.007	4.3
	Int 32	0.1	0.03	3.3
	BFloat 16	--	0.11	--
	IEEE FP 16	0.4	0.16	2.5
	IEEE FP 32	0.9	0.38	2.4
×	Int 8	0.2	0.07	2.9
	Int 32	3.1	1.48	2.1
	BFloat 16	--	0.21	--
	IEEE FP 16	1.1	0.34	3.2
	IEEE FP 32	3.7	1.31	2.8
SRAM	8 KB SRAM	10	7.5	1.3
	32 KB SRAM	20	8.5	2.4
	1 MB SRAM <sup>1</sup>	100	14	7.1
GeoMean <sup>1</sup>		--	--	2.6
DRAM		Circa 45 nm	Circa 7 nm	
	DDR3/4	1300 <sup>2</sup>	1300 <sup>2</sup>	1.0
	HBM2	--	250-450 <sup>2</sup>	--
	GDDR6	--	350-480 <sup>2</sup>	--

**Table 2. Energy per Operation: 45 nm [16] vs 7 nm. Memory is pJ per 64-bit access.**

Jouppi, Norman, *et al.*. “Ten Lessons From Three Generations Shaped Google’s TPUv4i : Industrial Product,” in *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, Valencia, Spain, 2021 pp. 1-14 at 3 (emphasis added). According to the above table, the energy per operation required by a bfloat16 multiplication circuit is under 20% of that required by an IEEE traditional-precision (*i.e.*, “FP 32”) multiplication circuit. The lower power requirements of low-precision bfloat16 multiplication circuits is a result of the fact that they include fewer transistors than traditional-precision multiplication circuits.

59. In knowingly incorporating Dr. Bates's patented computer architectures into the accused TPUs, Google reaps the very same benefits that were predicted by Dr. Bates in his patent application more than 10 years ago. As predicted by Dr. Bates in the '775 patent:

PEs implemented according to certain embodiments of the present invention may be relatively small for PEs that can do arithmetic. This means that there are many PEs per unit of resource (e.g., transistor, area, volume), which in turn means 40 that there is a large amount of arithmetic computational power per unit of resource. This enables larger problems to be solved with a given amount of resource than does traditional computer designs. For instance, a digital embodiment of the present invention built as a large silicon chip fabricated with 45 current state of the art technology might perform tens of thousand of arithmetic operations per cycle, as opposed to hundreds in a conventional GPU or a handful in a conventional multicore CPU. These ratios reflect an architectural advantage of embodiments of the present invention that 50 should persist as fabrication technology continues to improve, even as we reach nanotechnology or other implementations for digital and analog computing.

60. As a result of Google's IPR petitions and activities described above, including its monitoring of Singular's patent applications and patents, Google knew of application serial number 16/882,694 which led to the '775 patent since, at the latest, October 30, 2020 when Google identified the application in, *inter alia*, its Petition for *Inter Partes* Review in IPR2021-00154. Before making such identification, counsel for Google reviewed application serial number 16/882,694.

61. Google's infringement of the '775 patent will continue unless and until Google is enjoined.

62. As a result of Google's infringement of the '775 patent, Singular has been irreparably harmed and suffered damages in an amount to be determined at trial.

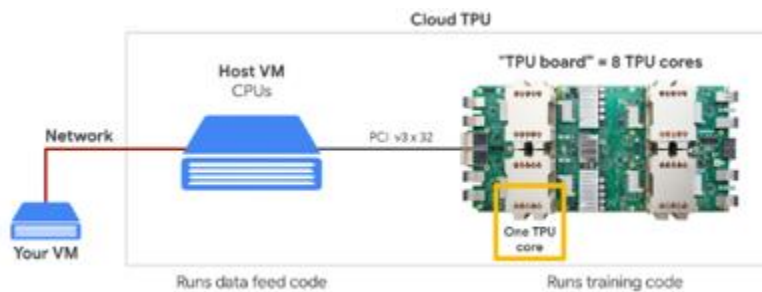
## **COUNT II** **INFRINGEMENT OF THE '616 PATENT**

63. Paragraphs 1-62 above are incorporated herein by reference.



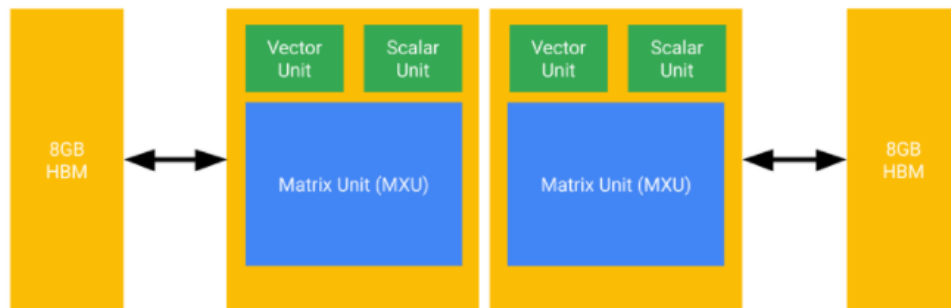
64. Google has directly infringed, and continues to directly infringe, literally and/or by the doctrine of equivalents, at least claim 10 of the '616 patent by making, testing, using, offering for sale, selling and/or importing into the United States the accused TPUs that are used inside Google's existing data centers.

65. According to Google's own published documents, each of the accused TPUs is a computer system comprising one host computer, namely a Host VM CPU ("TPU host"), and at least one TPU board. The TPU host is connected to each TPU board, and each TPU board in turn comprises one or more TPU computing chips ("TPU chips"). Each TPU chip in turn comprises a plurality of TPU cores. Each TPU board is connected to the TPU host for loading and preprocessing data for feeding into the TPU cores. *See* <https://cloud.google.com/tpu/docs/>:

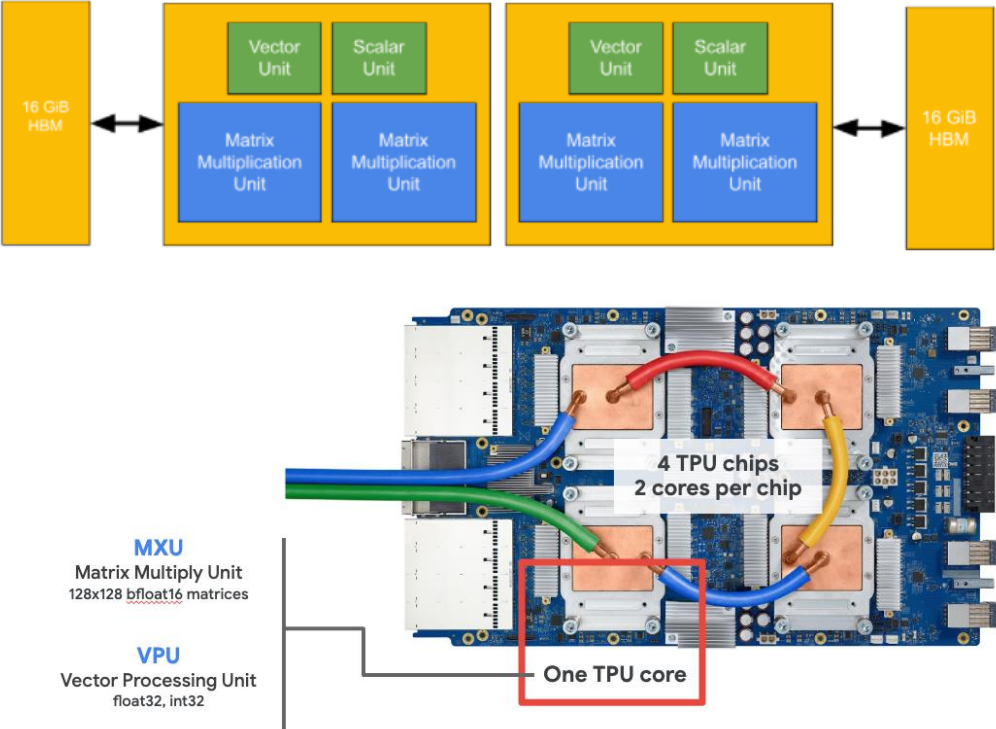


66. According to Google's publications, each TPU core comprises a Matrix Multiply Unit array ("MXU array") that performs matrix multiplication operations, a Vector Unit (also known as a Vector Processing Unit, or "VPU") and a Scalar Unit.

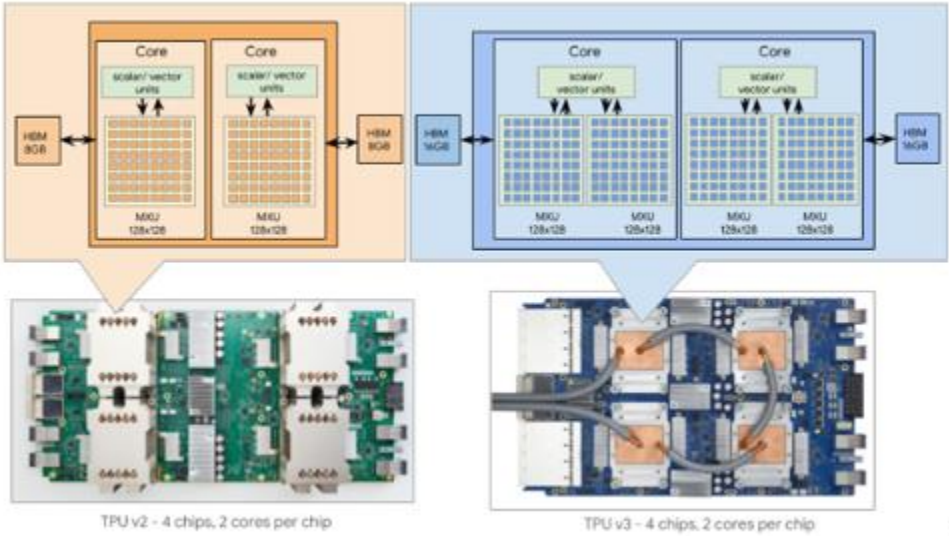
TPU v2:



TPU v3:

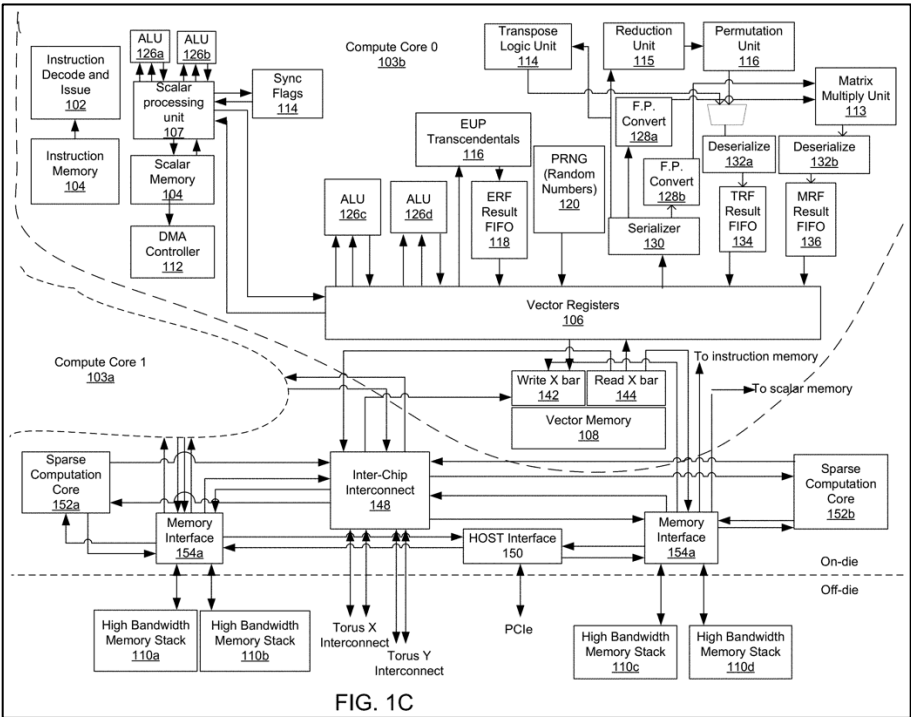


67. Each of the aforementioned TPU chips comprises at least one processing element array, as evidenced by, *e.g.*, the following excerpts from Google’s own publications:

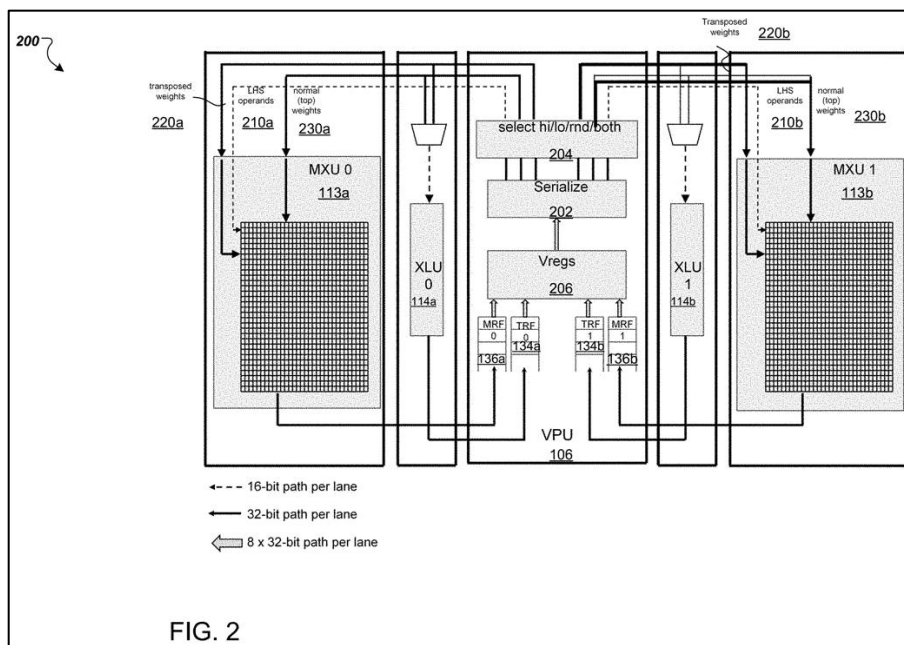


Google’s fourth-generation TPU ASIC offers more than double the matrix multiplication TFLOPs of TPU v3, a significant boost in memory bandwidth, and advances in interconnect technology. Google’s TPU v4 MLPerf submissions take advantage of these new hardware features with complementary compiler and modeling advances. The results demonstrate an average improvement of 2.7 times over TPU v3 performance at a similar scale in the last MLPerf Training competition.

TPU v4 double the matrix multiplication TFLOPs and offers a significant boost in memory bandwidth from new interconnect technology.



Google ’269 Patent, Fig. 1C (showing a “neural network processing system”)

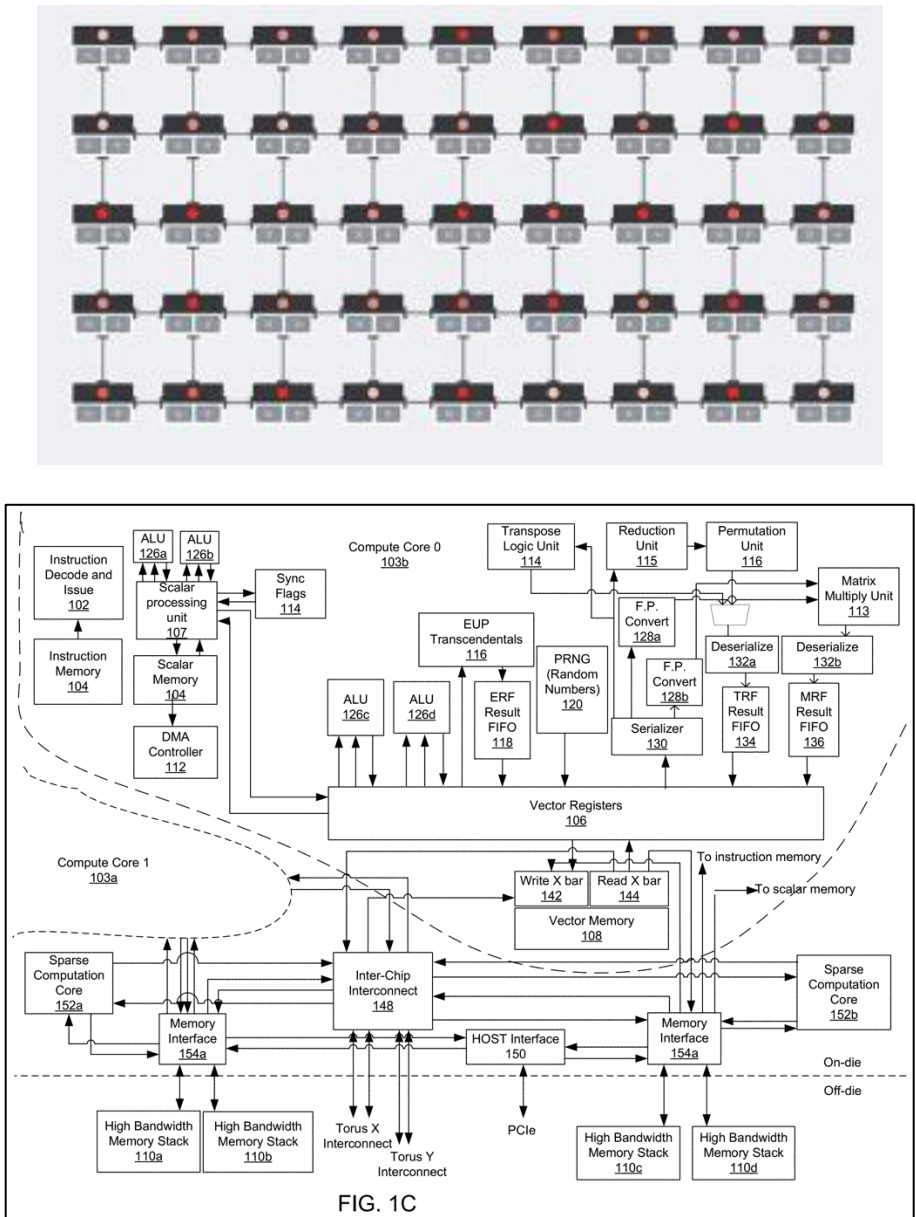


**Google '269 Patent, Fig. 2 (showing a “two-dimensional systolic array”)**

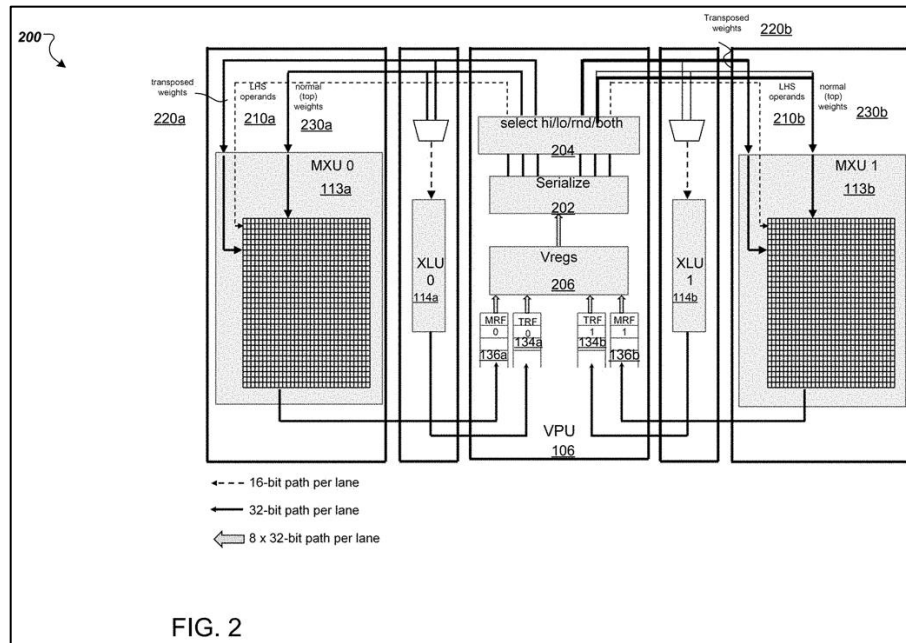
68. As published by Google, each processing element arrays comprises 128x128 processing elements, for a total of 16,384 processing elements per array. Thus, the accused TPUs each comprise more than 5,000 processing elements. *See* <https://cloud.google.com/tpu/docs/beginners-guide>:

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

69. According to Google’s publications, at least a first subset of the aforementioned processing elements are positioned at an edge of the aforementioned processing element array, and a second subset the aforementioned processing elements are positioned in the interior of the aforementioned processing element array.



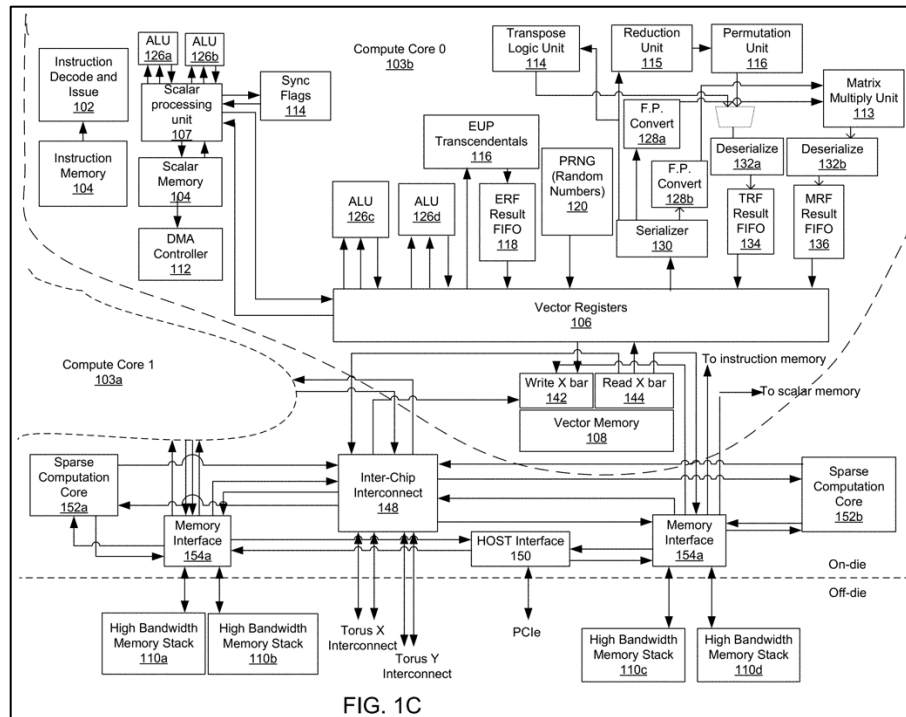
Google '269 Patent, Fig. 1C (showing a “neural network processing system”)



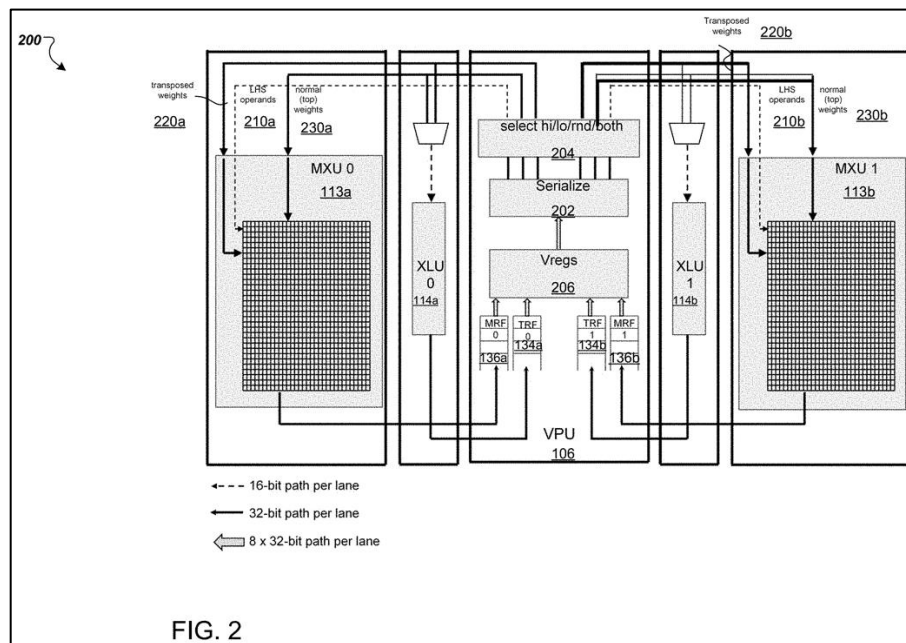
**Google '269 Patent, Fig. 2 (showing a “two-dimensional systolic array”)**

70. As shown in Google’s published documents, including the Google ’269 patent (excerpted below), each of the aforementioned TPU chips comprises at least one input-output unit (“TPU input-output unit”) connected to the aforementioned first subset of processing elements positioned at the edge of the aforementioned processing element array:





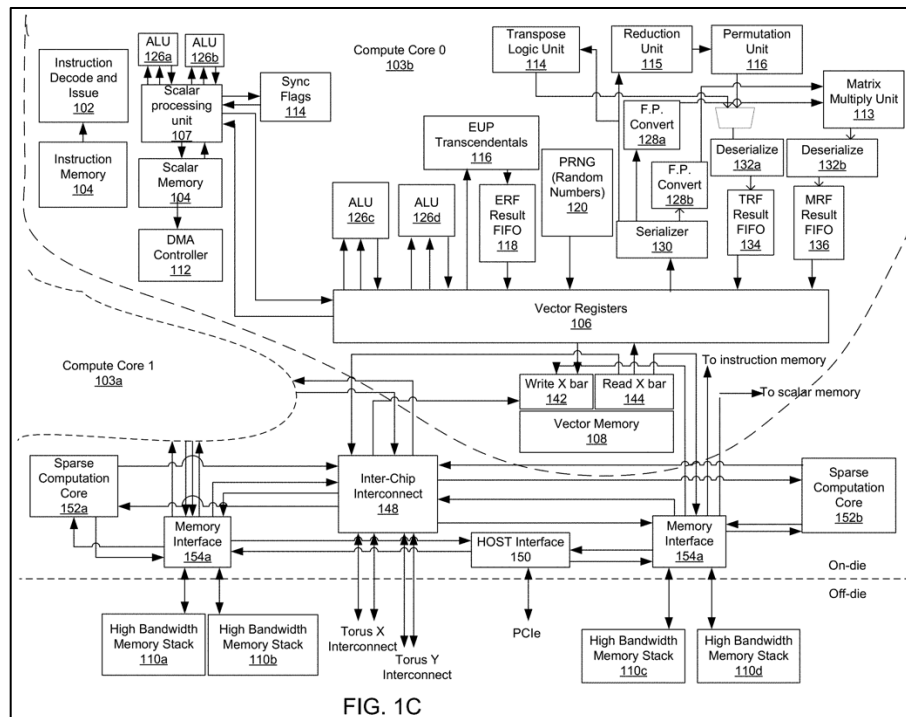
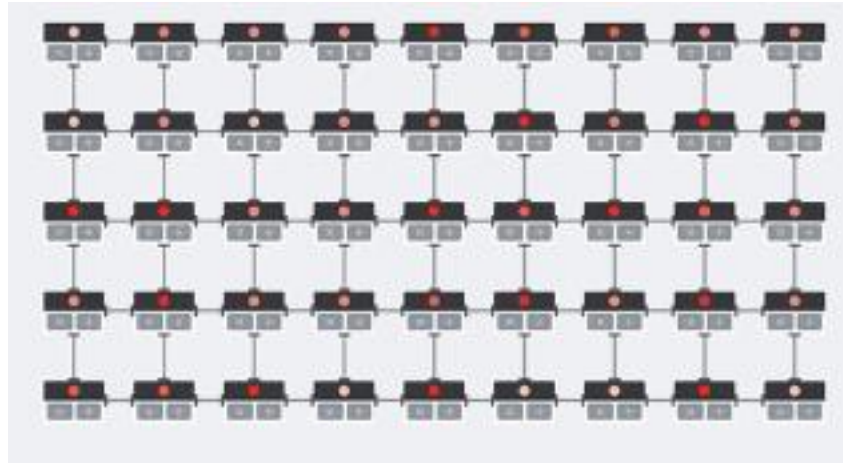
Google '269 Patent, Fig. 1C (showing a “neural network processing system”)



Google '269 Patent, Fig. 2 (showing a “two-dimensional systolic array”)

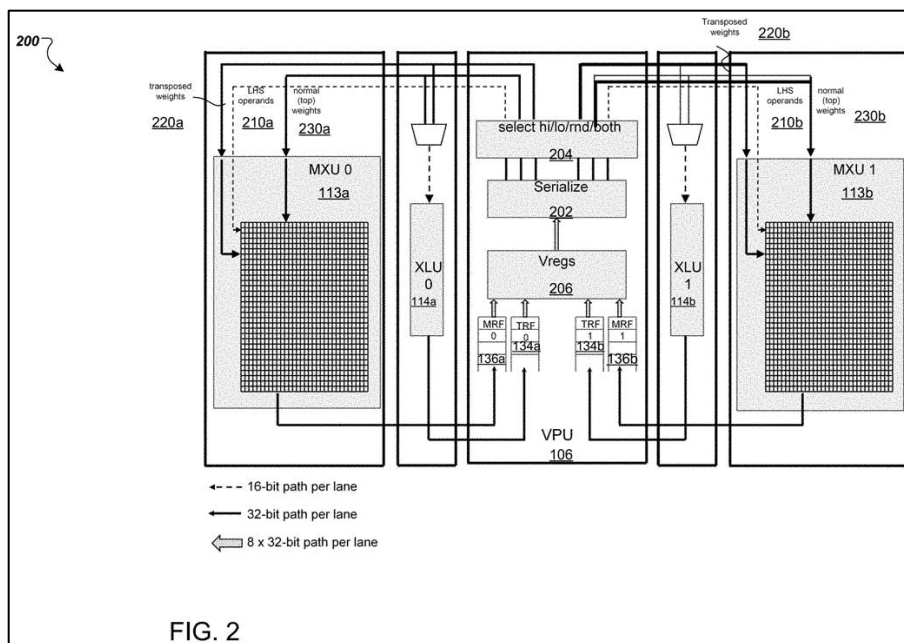
71. As shown in Google’s published documents, each of the aforementioned TPU chips comprises a plurality of processing element connections, each of which connects one of the aforementioned processing elements with at least one other of the aforementioned processing

elements. Each of the aforementioned processing elements is connected to at least another of the other processing elements by at least one of the processing element connections.

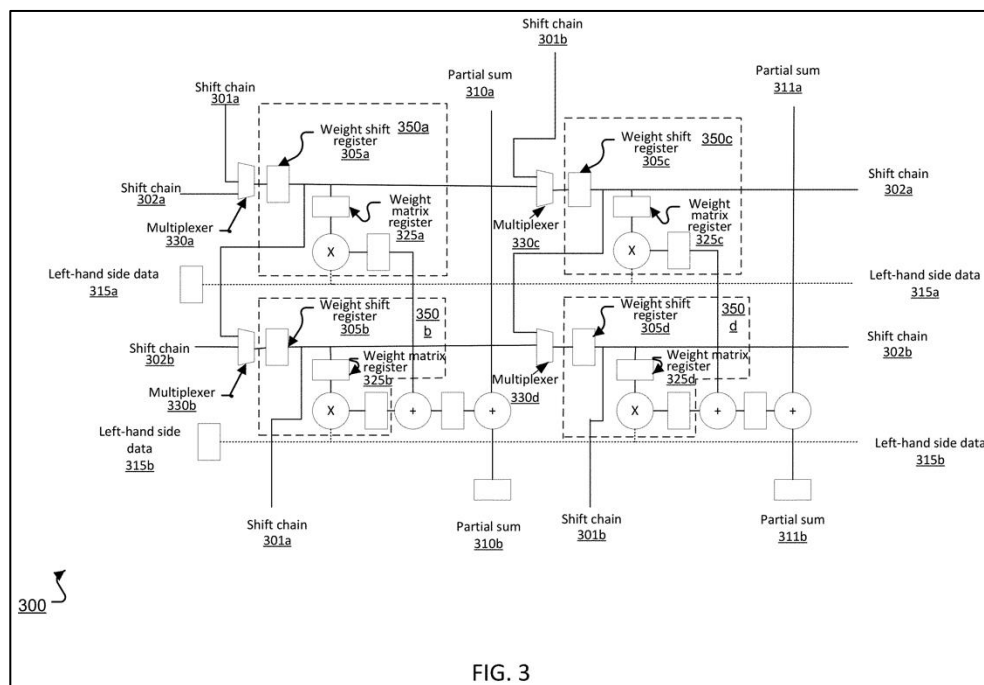


Google '269 Patent, Fig. 1C (showing a “neural network processing system”)





**Google '269 Patent, Fig. 2 (showing a “two-dimensional systolic array”)**



**Google '269 Patent, Fig. 3 (showing a “multi-cell inside a systolic array”)**

72. As shown in Google's published documents, each of the aforementioned TPU chips comprises a plurality of memory units. Each of the aforementioned processing elements of the aforementioned TPU chips is associated with a corresponding one of the aforementioned

plurality of memory units, and each such memory unit is local to its associated one of the aforementioned processing elements. *See, e.g.*, <https://cloud.google.com/tpu/docs/beginners-guide> (“the TPU loads the parameters from memory into the matrix of multipliers and adders”).

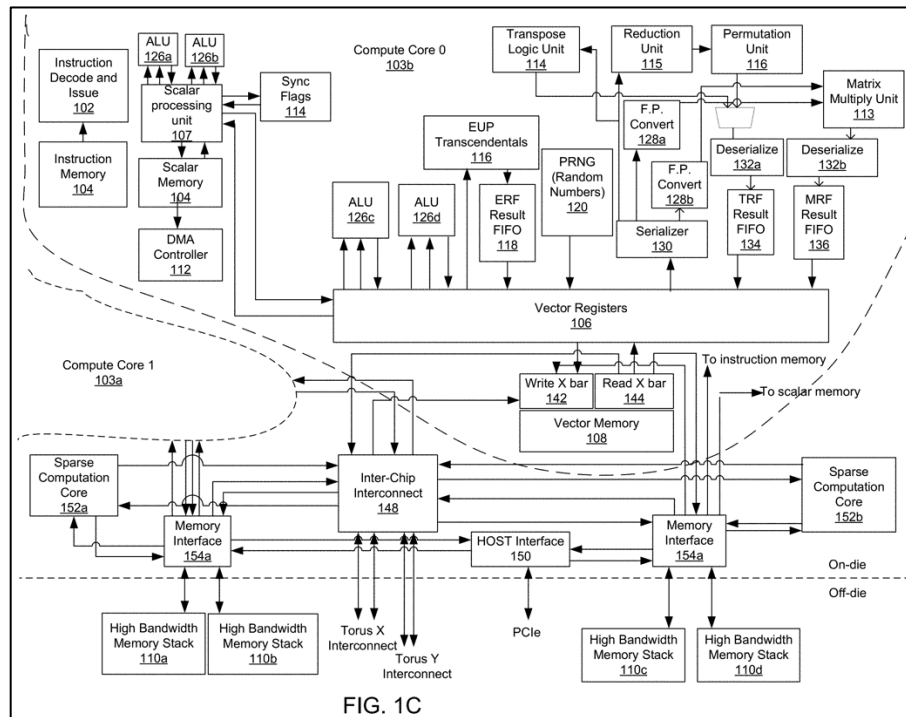


FIG. 1C

Google '269 Patent, Fig. 1C (showing a “neural network processing system”)

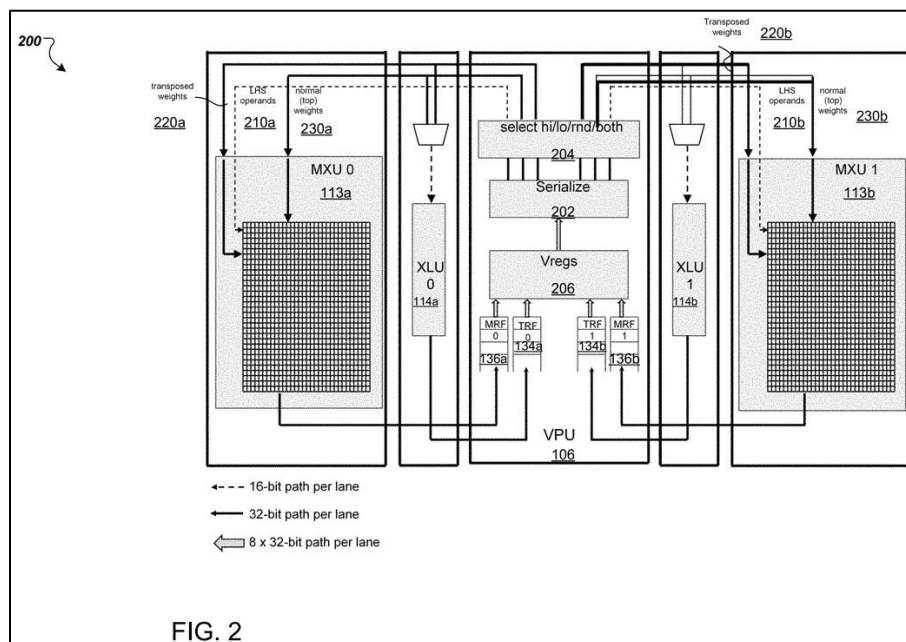
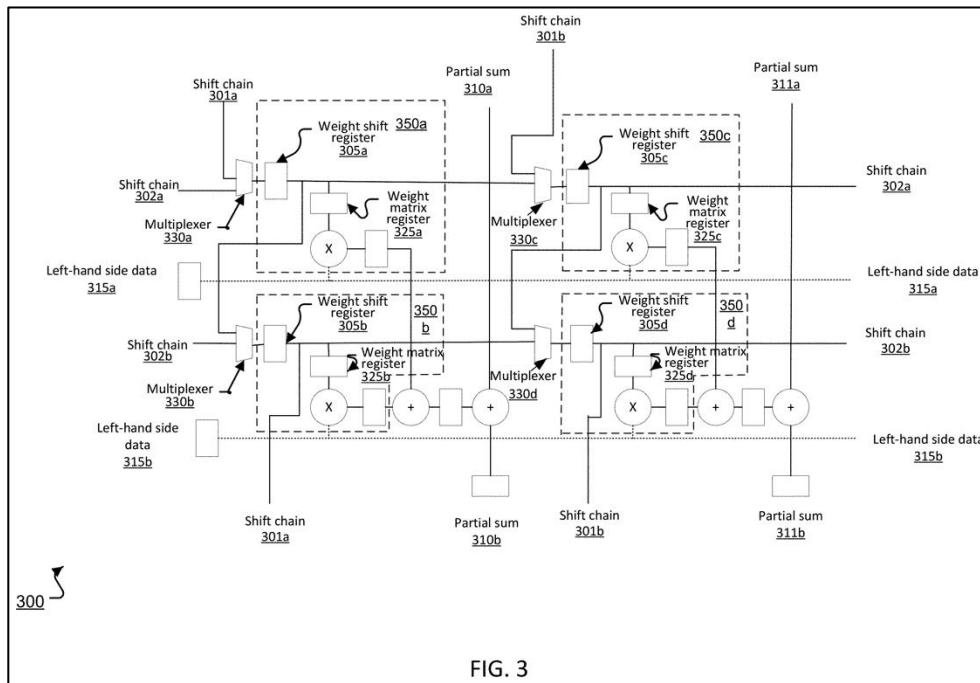


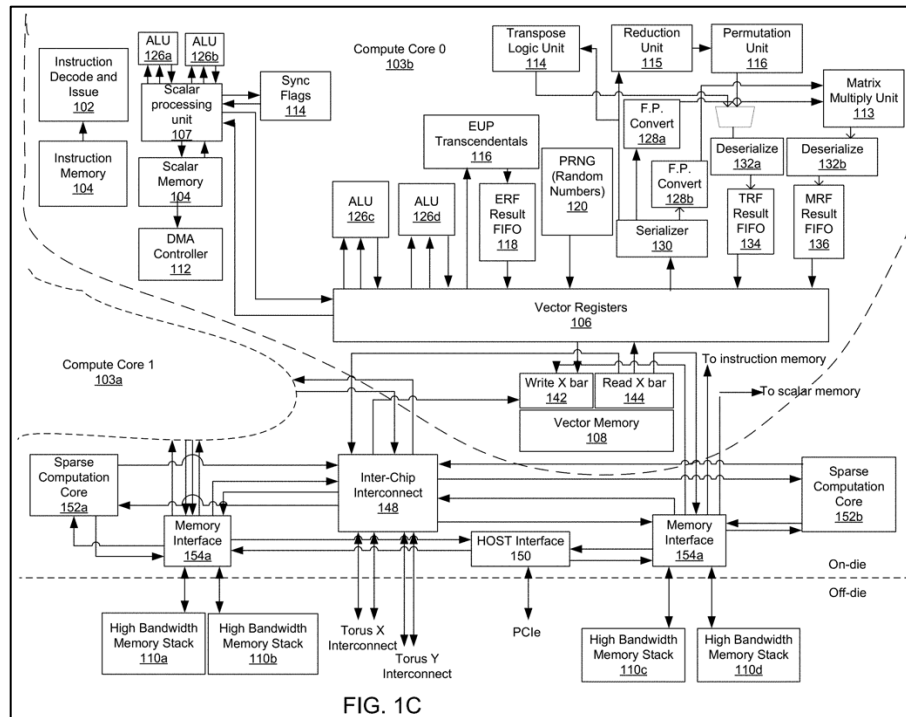
FIG. 2

Google '269 Patent, Fig. 2 (showing a “two-dimensional systolic array”)

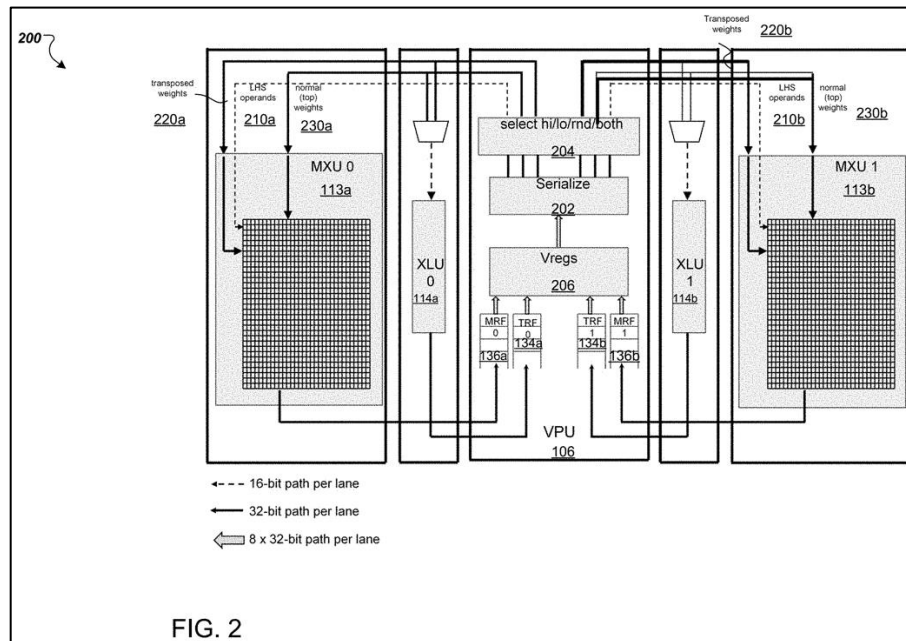


Google '269 Patent, Fig. 3 (showing a “multi-cell inside a systolic array”)

73. Google’s own publications further show that each of the aforementioned processing elements in the aforementioned TPU chips has positioned therein one arithmetic unit (“MXU arithmetic unit”).



**Google '269 Patent, Fig. 1C (showing a “neural network processing system”)**



**Google '269 Patent, Fig. 2 (showing a “two-dimensional systolic array”)**

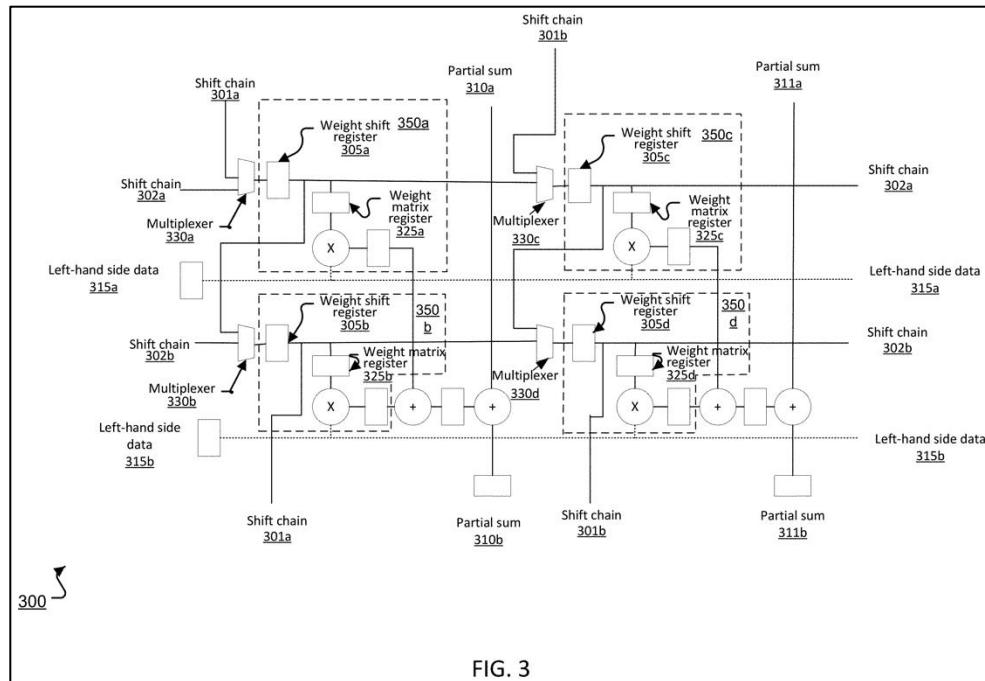
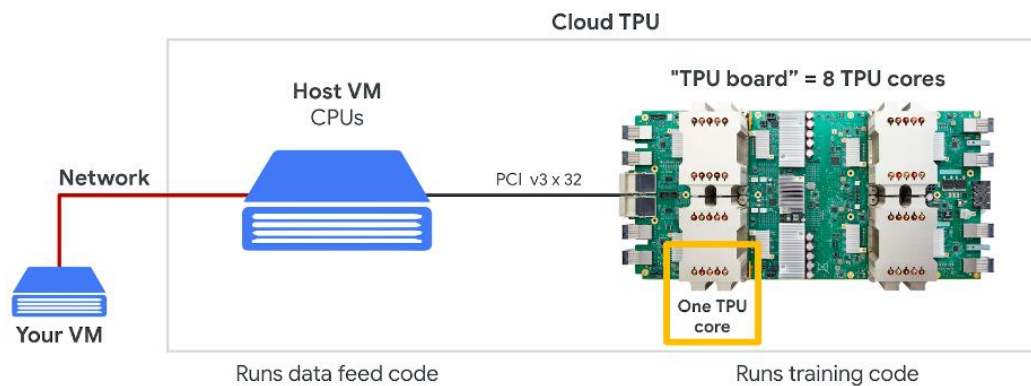
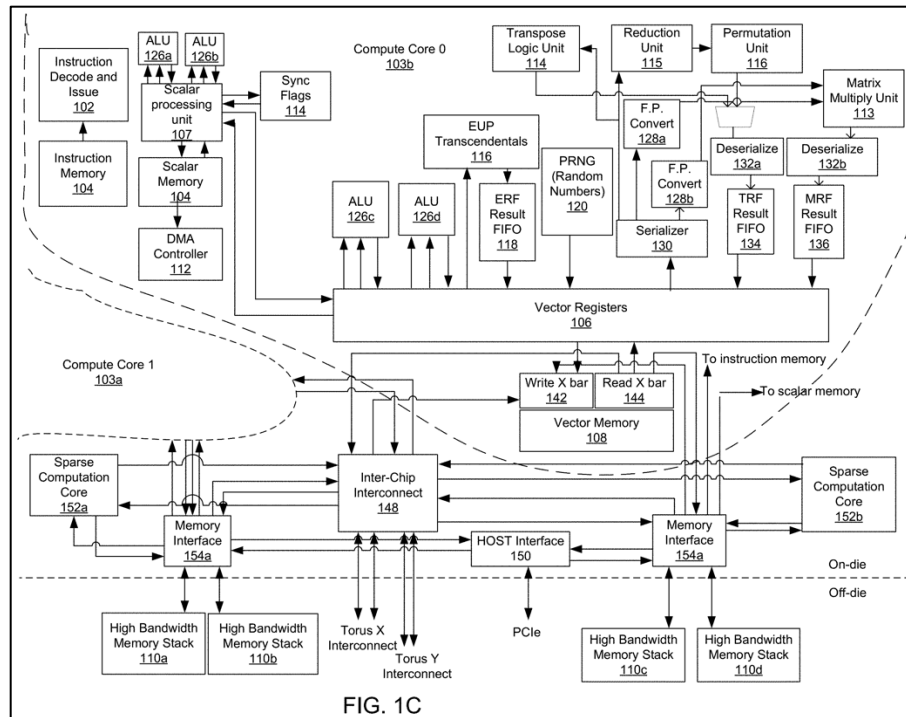


FIG. 3

Google '269 Patent, Fig. 3 (showing a “multi-cell inside a systolic array”)

74. Each of the accused TPU devices comprises a host connection that at least partially connecting the aforementioned TPU input-output unit with the aforementioned TPU host, according to Google’s own published documentation at <https://codelabs.developers.google.com/codelabs/keras-flowers-data#2>:





**Google '269 Patent, Fig. 1C (showing a “neural network processing system”)**

75. According to Google’s published documents, each of the aforementioned MXU arithmetic units comprises a corresponding multiplier circuit (“MXU multiplier circuit”). Each MXU multiplier circuit is adapted to receive as inputs two floating point values having a bfloat16 format.



Cloud TPU

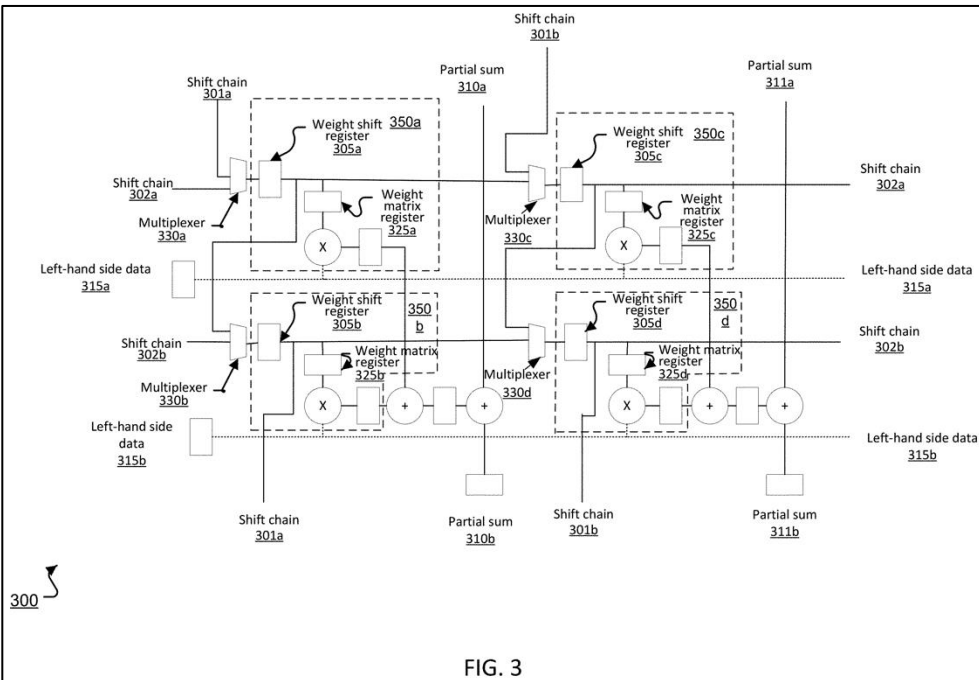
## System Architecture

Each TPU core has scalar, vector, and matrix units (MXU). The MXU provides the bulk of the compute power in a TPU chip. Each MXU is capable of performing 16K multiply-accumulate operations in each cycle. While the MXU inputs and outputs are 32-bit floating point values, the MXU performs multiplies at reduced **bfloat16** precision. Bfloat16 is a 16-bit floating point representation that provides better training and model accuracy than the IEEE **half-precision** representation.

Cloud TPU v2 and Cloud TPU v3 primarily use bfloat16 in the matrix multiplication unit (MXU), a 128 x 128 systolic array. There are two MXUs per TPUv3 chip and multiple TPU chips per Cloud TPU system. Collectively, these MXUs deliver the majority of the total system FLOPS. Each MXU takes inputs in FP32 format but then automatically converts them to bfloat16 before calculation. (A TPU can perform FP32 multiplications via multiple iterations of the MXU.) Inside the MXU, multiplications are performed in bfloat16 format, while accumulations are performed in full FP32 precision.

## Choosing bfloat16

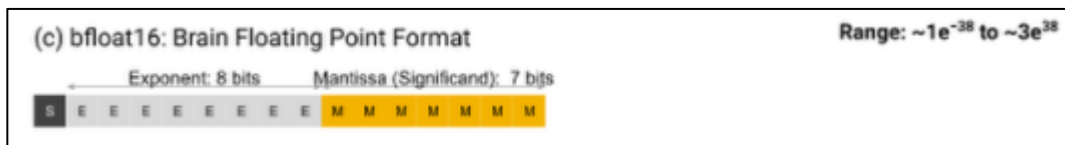
Our hardware teams chose bfloat16 for Cloud TPUs to improve hardware efficiency while maintaining the ability to train accurate deep learning models, all with minimal switching costs from FP32. The physical size of a hardware multiplier scales with the *square* of the mantissa width. With fewer mantissa bits than FP16, the bfloat16 multipliers are about half the size in silicon of a typical FP16 multiplier, and they are *eight times* smaller than an FP32 multiplier!



Google '269 Patent, Fig. 3 (showing a “multi-cell inside a systolic array”)



76. According to Google's own publications, the bfloat16 format characterizing the input received by each MXU multiplier circuit of the accused TPUs, has a sign bit, 8 exponent bits and 7 mantissa bits:



77. As described by Google above, the bfloat16 format utilizes a binary mantissa of width that is no more than 11 bits and a binary exponent of width that is at least 6 bits. Google copied the idea of a computer device having an array of processing elements that perform floating-point arithmetic using such a number format, from Dr. Bates.

78. Each of the aforementioned VPUs in the aforementioned TPU chips, comprises a plurality of processing elements that each comprises a multiplier circuit (“VPU multiplier circuit”) that is adapted to receive as inputs two floating point values each of a width that is at least 32 bits wide. *See, e.g.*, <https://codelabs.developers.google.com/codelabs/keras-flowers-data/#2> (“The VPU handles float32 and int32 computations”)

The vector processor consists of a 2-dimensional array of  
 40 vector processing units, i.e.,  $128 \times 8$ , which all execute the same instruction in a single instruction, multiple-data (SIMD) manner. The vector processor has lanes and sublanes, i.e., 128 lanes and 8 sublanes. Within the lane, the vector units communicate with each other through load and  
 45 store instructions. Each vector unit can access one 4-byte value at a time. Vector units that do not belong to the same lane cannot communicate directly. These vector units must use the reduction/permutation unit which is described below.

The computational unit includes vector registers, i.e., 32  
 50 vector registers, in a vector processing unit (106) that can be used for both floating point operations and integer operations. The computational unit includes two arithmetic logic units (ALUs) (126c-d) to perform computations. One ALU (126c) performs floating point addition and the other ALU  
 55 (126d) performs floating point multiplication. Both ALUs

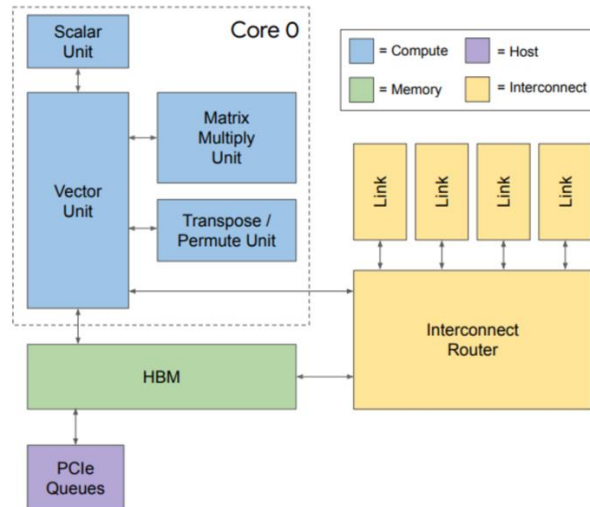
**Google '269 Patent, 6:39-55**

79. As Google’s publications show, in each of the accused TPU chips, there are at least 100 more processing elements that each comprise an MXU multiplier circuit (which each takes as inputs a pair of bfloat16 floating point numbers), than processing elements that each comprise a VPU multiplier circuit (which each takes as inputs a pair of floating point numbers that are each at least 32 bits wide). See <https://blog.intel.com/2018/01/hardware-for-deep-learning-part-4-asic-96a542fe6a81>. This is also confirmed by the Google ’269 patent, which states that, “[t]he vector processor consists of a 2-dimensional array of vector processing units, *i.e.*, 128×8, which all execute the same instruction in a single instruction.” Google ’269 patent, 6:39-41; *see also id.* at 6:61 (explaining that the vector processor “performs multiple, *i.e.*, 1024, operations in one clock cycle”), and also states that “[e]ach MXU may have 128 rows and 128 columns.” Google ’269 Patent at 8:59. This means that in each of the accused TPU chips that have an MXU array for each VPU, there are 16,384 MXU multiplier circuits (each of which takes as inputs a pair of bfloat16 floating point numbers), for every 1,024 VPU multiplier circuits (which each takes as inputs a pair of floating point numbers that are each at least 32 bits wide), meaning in turn there are at least 100 more of the former than the latter. The difference is even greater in TPU chips in which there are multiple MXU arrays for each VPU.

TPU v2 is connected to the host by PCIe Gen3 x32.

The MXU provides the bulk of computing power in a TPU chip. Each MXU is capable of performing 16K (128x128) multiply-accumulate operations in each cycle at reduced **bfloat16 (BF16)** precision ([more on bfloat16 and other formats here](#)). It is supported by a vector processing unit that performs all the other computations in typical training workloads.

Matrix multiplications are performed with BF16 inputs but all accumulations happen in FP32 so the resulting matrix is FP32. All other computations are in FP32 except for results going directly to an MXU input, which are converted to BF16.



The *Scalar Unit* fetches VLIW (*Very Long Instruction Word*) instructions from the core's on-chip, software-managed Instruction Memory (Imem), executes scalar operations using a 4K 32-bit scalar data memory (Smem) and 32 32-bit scalar registers (Sregs), and forwards vector instructions to the Vector Unit. The 322-bit VLIW instruction can launch eight operations: two scalar, two vector ALU, vector load and store, and a pair of slots that queue data to and from the matrix multiply and transpose units. The XLA compiler schedules loading Imem via independent overlays of code, as unlike conventional CPUs, there is no instruction cache.

The *Vector Unit* performs vector operations using a large on-chip *vector memory* (Vmem) with 32K 128 x 32-bit elements (16MB), and 32 2D *vector registers* (Vregs) each containing 128 x 8 32-bit elements (4 KB). The Vector Unit streams data to and from the MXU through decoupling FIFOs. The Vector Unit collects and distributes data to Vmem via *data-level parallelism* (2D matrix and vector functional units) and *instruction-level parallelism* (8 operations per instruction).

30 The chip stores data in high bandwidth memory (156c-d), reads the data in and out of vector memory (108), and processes the data. The compute core (103b) itself includes a vector memory (108) that is on-chip S-RAM which is divided into two dimensions. The vector memory has  
 35 address space in which addresses hold floating point numbers, i.e., 128 numbers that are each 32-bits. The compute core (103b) also includes a computational unit that computes values and a scalar unit that controls the computational unit.

The vector processor consists of a 2-dimensional array of  
 40 vector processing units, i.e., 128×8, which all execute the same instruction in a single instruction, multiple-data (SIMD) manner. The vector processor has lanes and sublanes, i.e., 128 lanes and 8 sublanes. Within the lane, the vector units communicate with each other through load and  
 45 store instructions. Each vector unit can access one 4-byte value at a time. Vector units that do not belong to the same lane cannot communicate directly. These vector units must use the reduction/permutation unit which is described below.

The computational unit includes vector registers, i.e., 32  
 50 vector registers, in a vector processing unit (106) that can be used for both floating point operations and integer operations. The computational unit includes two arithmetic logic units (ALUs) (126c-d) to perform computations. One ALU (126c) performs floating point addition and the other ALU  
 55 (126d) performs floating point multiplication. Both ALUs

**Google '269 Patent, 6:30-55**

80. According to documents published by Google, each of the accused TPUs perform inference and training. TPU v2 was the first version of Google's TPU products that performed both inference and training. *See* Jouppi, *et al.* "A Domain-Specific Supercomputer for Training Deep Neural Networks," p. 67

We're excited to announce that our second-generation Tensor Processing Units (TPUs) are coming to [Google Cloud](#) to accelerate a wide range of machine learning workloads, including both training and inference. We call them [Cloud TPUs](#), and they will initially be available via [Google Compute Engine](#).

81. For example, as published by Google, the accused TPUs perform inference for users of Google Photos to analyze the similarity of a user-inputted image, to other images. These other images are searched by Google to estimate whether or not the user-inputted image is similar to any of the other images (*e.g.*, where the subject of the user-inputted image is a face, and subject of each of the other images are also faces, and the former is compared to each of the latter images so as to determine whether or not any of the other images match the user-inputted image):

Face grouping occurs in 3 steps:

1. We detect whether any photo has a face in it.
2. If the face grouping feature is turned on, algorithmic models are used to predict the similarity of different images and estimate whether 2 images represent the same face.
3. Photos that are likely to represent the same face are grouped together. You can always remove a photo from a group if you think it's in the wrong group.

When face grouping is on, Google Photos may also include photos in a particular group based on other characteristics. This includes photos being taken close together in time and detecting that a person is wearing the same clothing across photos when a face is not visible.

See <https://support.google.com/photos/>. This means the TPU host (which as shown above is used for loading and preprocessing data for feeding into the TPU cores) provides instructions to a TPU chip, that cause the TPU chip to perform an operation whose output is used to identify an image from a plurality of images to be searched that is similar to a user-inputted image. The accused TPUs perform a similar predictive image comparison functionality for users of, *inter alia*, Google Lens (see <https://lens.google.com/howlensworks/>) and Google Images (see <https://images.google.com>). Google also offers its reverse image technology performed by the accused TPUs to third parties such as The New York Times and Box with instructions on how to use the technology with the intent that third parties use the technology in such an infringing manner. See <https://cloud.google.com/vision>.

82. Google's infringement of the '616 patent is and has been willful.

83. Less than two years after the filing of his provisional application in June 2009, Dr. Bates and Google executed a Non-Disclosure Agreement (“NDA”) prepared by Google. *See* Amended Answer (Dkt. No. 57) in case No. 1:19-cv-12551 (D. Mass.) (“Am. Ans.”), ¶ 17.

84. On November 3, 2010, Joseph Bates forwarded a document titled “COMPUTING 10,000X MORE EFFICIENTLY” by email to Astro Teller at the email address [astroteller@google.com](mailto:astroteller@google.com).

85. On November 3, 2010, Joseph Bates discussed Singular’s technology by telephone with Astro Teller while Astro Teller was in Massachusetts for, *inter alia*, a meeting with Joseph Bates at the Massachusetts Institute of Technology Media Lab.

86. On December 9, 2010, Joseph Bates forwarded a document titled “COMPUTING 10,000X MORE EFFICIENTLY” by email to Sebastian Thrun at the email address [thrun@google.com](mailto:thrun@google.com) with copies to Astro Teller and Sergio Gandara.

87. On December 9, 10 and 21, 2010, Joseph Bates discussed Singular’s technology by telephone with Astro Teller.

88. After receiving the document titled “COMPUTING 10,000X MORE EFFICIENTLY,” Astro Teller discussed Singular’s technology with Larry Page and/or Sergey Brin.

89. On January 28, 2011, Joseph Bates discussed Singular’s technology by telephone with Astro Teller.

90. On June 9, 2011, Joseph Bates discussed Singular’s technology with Astro Teller, Sebastian Thrun and Megan Smith at a meeting at the Massachusetts Institute of Technology Media Lab.

91. On June 21, 2011, Joseph Bates forwarded a document titled “APPLICATIONS / MARKETS / AND DEALS” by email to Astro Teller at the email address [astroteller@google.com](mailto:astroteller@google.com).

92. On June 22, 2011, Joseph Bates forwarded a document titled “APPLICATIONS / MARKETS” by email to Astro Teller at the email address [astroteller@google.com](mailto:astroteller@google.com).

93. On June 24, 2011, Joseph Bates met with Astro Teller, Johnny Chen, and others from Google to discuss Singular’s technology.

94. On June 22, 2011, Joseph Bates forwarded a document titled “SINGULAR COMPUTING” by email to Astro Teller at the email address [astroteller@google.com](mailto:astroteller@google.com).

95. On June 22, 2011, Joseph Bates forwarded a document titled “APPLICATIONS / MARKETS” by email to Astro Teller at the email address [astroteller@google.com](mailto:astroteller@google.com).

96. On September 17, 2013, Joseph Bates met with Google’s Jeffrey Dean, Quoc Le and others at Google. Pursuant to the NDA between Google and Singular, a slide presentation titled “MULTI-MILLION CORE PROCESSORS AND THEIR APPLICATIONS” was loaded onto a Google laptop from which Dr. Bates displayed the slides to Jeffrey Dean and Quoc Le. Thereafter, on September 17, 2013, Jeffrey Dean emailed Dr. Bates stating: “[a] few folks here are interested in seeing if we can train neural nets with various kinds of computational inaccuracies.”

97. On January 22, 2014, Dr. Bates emailed Jeffrey Dean referencing “Singular’s hardware, software, patents, experience, etc.” In an email response dated January 23, 2014, Jeffrey Dean stated, *inter alia*, that he had “passed this info along to two people I think are most relevant within Google.”



98. On or around January 24, 2014, Dr. Bates forwarded a presentation titled “MANY-MILLION CORE PROCESSORS AND THEIR APPLICATIONS” to Nanette Boden at the email address nanboden@google.com with copies to Jeffrey Dean and Norm Jouppi. The presentation stated that it was “Confidential, per Google/Singular MNDA, March 2011.” In the presentation, Dr. Bates warned Google that Singular had patent protection relating to the disclosed Singular technology.

99. On February 2, 2017, Dr. Bates met with Astro Teller, Tammo Spalink and others at Google in Mountain View, California to make a presentation and demonstration of Singular’s patented technology. On February 27, 2017, James Laudon asked Dr. Bates for a set of the presentation slides. On March 1, 2017, Dr. Bates sent a copy of the slides, titled “APPROXIMATE COMPUTING, EMBEDDED AI, BILLION CORE SYSTEMS,” to James Laudon.

100. On February 20, 2017, Obi Felten of Google’s X team informed Dr. Bates by email that “Catherine Tornabene from the X IP legal team . . . will review your patent family.”

101. On March 1, 2017, Jenn Wall, then a commercial lawyer in Google’s X team, forwarded to Dr. Bates by email a draft Mutual Confidentiality and Non-Disclosure Agreement (“draft MNDA”). Paragraph 8 of the draft MNDA contained the following language:



The Company [Singular] waives any right to allege willful infringement based on notice to or knowledge by Google of any patent identified by the Company to Google (a) under this Agreement or (b) in any communication related to the Transaction prior to the effective date of this Agreement.





102. Dr. Bates did not sign Google’s draft MNDA that was forwarded on March 1, 2017.

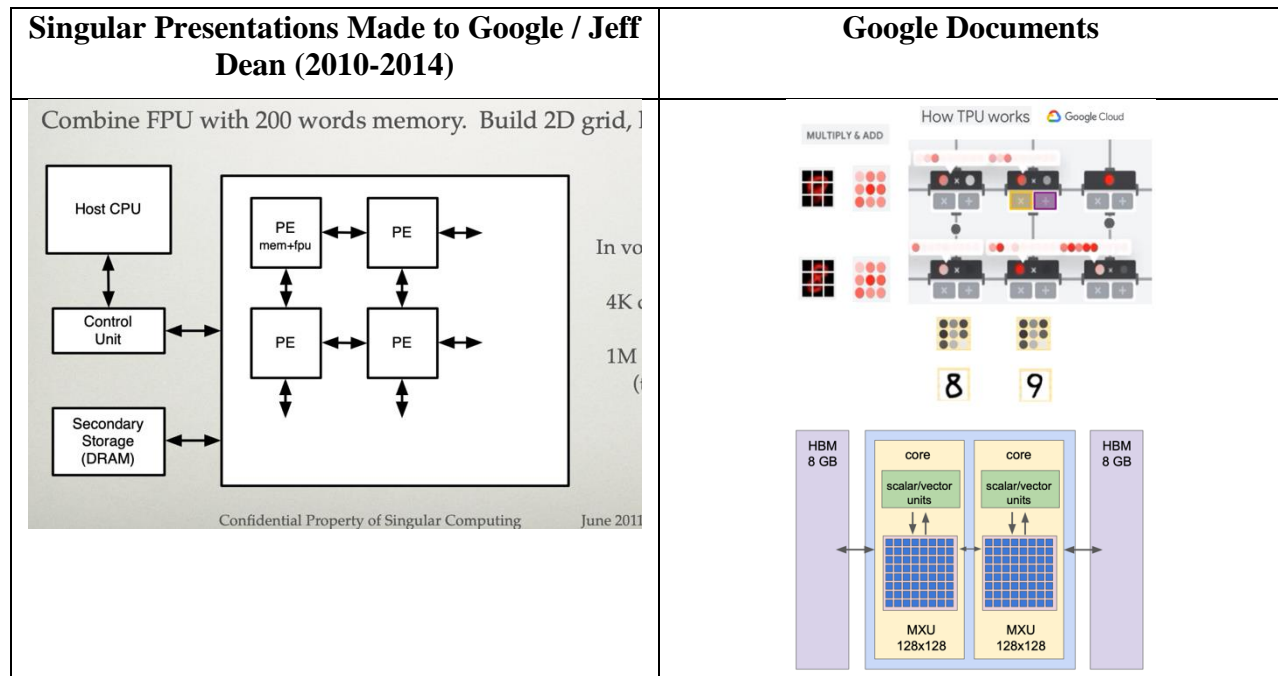
103. During the course of these meetings, Dr. Bates discussed his computer theories with certain Google employees. *See* Am. Ans., ¶ 19. Dr. Bates also advised Google that his

disclosed computer architecture and S1 prototype were patent-protected. For example, in a presentation titled “Approximate Computing, Embedded AI, Billion Core Systems,” Dr. Bates informed representatives of Google in February 2017 that Singular had “broad patents granted in U.S. and internationally.”

104. Following Dr. Bates’s disclosure of his inventions, Google copied and adopted Dr. Bates’s patented inventions, incorporating the disclosed technology into the accused TPUs installed in Google’s data centers. This is apparent from a comparison of Dr. Bates’s patented architecture and that of the accused TPUs. It is also apparent from an exemplary comparison of the disclosures made in writing by Dr. Bates to Google with the properties and features that Google later adopted in its accused TPUs. For example:

Singular Presentations Made to Google / Jeff Dean (2010-2014)	Google Documents
	<p>Google Publication of TPUv2 (2017) and TPUv3 (2018)</p>  <p>Machine Learning for Systems and Systems for Machine Learning</p> <p>Jeff Dean Google Brain team <a href="http://g.co/brain">g.co/brain</a></p> <p>Presenting the work of <b>many</b> people at Google</p>

Singular Presentations Made to Google / Jeff Dean (2010-2014)	Google Documents
<div data-bbox="282 279 742 575"><p>(SINGULAR'S)</p><p><b>APPROXIMATE COMPUTING</b></p><p>A traditional massively parallel machine, with floating point arithmetic that is "99% correct"</p><p>(e.g. <math>1.0 + 1.0 = 1.98 \dots 2.02</math>)</p></div> <div data-bbox="250 583 774 795"><ul style="list-style-type: none"><li>• <b>Surprise #1:</b> Arithmetic circuit can be <u>unexpectedly</u> small</li></ul><div data-bbox="347 659 761 789"><div>Standard FPU ~500,000 transistors</div><div>~100x smaller standard deterministic digital cmos</div><div>+ - * / sqrt one cycle per op</div></div></div>	<div data-bbox="846 279 1092 615"><p>Special computation properties</p><p>reduced precision ok</p><p>about 1.2 = about 0.6 about 0.7</p><p>NOT</p><p>1.2104 = 0.6127 0.798912</p><p>handful of specific operations</p></div> <div data-bbox="1114 306 1417 720"><p>"We started to look at what we could do for these kinds of <u>deep learning models that could be more computationally efficient</u> and there are two really nice properties that deep learning models have. First, they are very tolerant of reduced precision... you don't need 6 or 7 digits of precision like you would in floating point computations or even more in double computations... you can build hardware that is only designed to accelerate low precision linear algebra, you're golden, and that enables you then really tailor the hardware to do only that,"</p></div>
<div data-bbox="297 856 719 911"><p><b>OTHER PROMISING DOMAINS</b> (IN PROGRESS - INITIAL EVIDENCE)</p></div> <div data-bbox="209 961 812 1257"><ul style="list-style-type: none"><li>• Vision: segmenting smooth objects (weak features, Hartmut/Joe intuition)</li><li>• Molecular dynamics, Protein folding (all-atom energy)</li><li>• Genomics (eg Smith-Waterman dynamic programming)</li><li>• Machine learning (neural nets, genetic algorithms with local crossover, local graphical models, simulated annealing)</li><li>• Speech recognition (HMMs, many concurrent voice streams, Dragon CTO)</li><li>• Neocortex sim (&gt;human, faster than realtime, supercomputer \$)</li></ul></div> <div data-bbox="354 1306 787 1320"><p>Confidential Property of Singular Computing June 2011 19</p></div>	<div data-bbox="846 846 1092 1257"><p>Machine Learning for Systems and Systems for Machine Learning</p><p>Jeff Dean Google Brain team <a href="http://g.co/brain">g.co/brain</a></p><p>Presenting the work of many people at Google</p></div> <div data-bbox="1114 865 1425 1354"><p>"Around the time of maybe 2011, 2012, when the Google Brain project that I co-founded was just getting started, we started to collaborate with . . . the speech recognition team [at Google] . . . and so we could tell that as <u>speech recognition</u> gets better people are going to use it more and more . . . and at the time, we had [sic] lots and lots of CPUs in our data center and if you look at how much computation that would be required if a hundred million of our users started to do that, that was actually kind of daunting and scary, we would have essentially double the computing footprint of Google just to support like a slightly better <u>speech recognition model</u>.</p></div>



105. Due to its monitoring of Singular's patents and applications, Google knew of the application for the '616 patent prior to the issuance of the patent on November 9, 2021. For example, Google's attorneys prepared and filed six petitions for *Inter Partes* Review ("IPR") of patents related to the '616 patent. In each of those petitions, Google identified numerous patents and applications related to the '616 patent, as well as application serial number 16/882,686 for the '616 patent.

106. Before identifying the application, Google reviewed application serial number 16/882,686.

107. Google has known and/or should have known of the '616 patent since its issuance or, at the latest, on or before Google's receipt of service of the original complaint in this case. Nonetheless, Google failed to cease its infringing activities or to seek a license to practice the inventions claimed in the patent. Alternatively, Google was willfully blind to application serial number 16/882,686 and to the issuance and its infringement of the '616 patent.

108. As set forth above, in an attempt to prevent Google from stealing Dr. Bates's inventions, Singular entered into an NDA with Google prior to Dr. Bates disclosing his inventions to representatives of Google. Notwithstanding the existence of the NDA, Google copied Dr. Bates's inventions. Since the issuance of the '616 patent, Google has done nothing to avoid infringing the '616 patent.

109. Further evidence of Google's nefarious conduct pertinent to this matter is Google's attempt to induce Dr. Bates into executing the MNDA in 2017. The MNDA included a provision, drafted by counsel for Google, whereby Dr. Bates would, *inter alia*, waive claims for willful infringement. At the time that Google attempted to induce Dr. Bates into executing the MNDA, Google knew or should have known that it had incorporated Dr. Bates's inventions previously disclosed by Dr. Bates to Google's representatives into the design of the TPU v2 and/or TPU v3 and that Singular had patents covering such inventions.

110. When Google learned, or should have learned, of the issuance of the '616 patent, Google should have ceased all manufacture, use, offering for sale and selling of the TPU v2 and TPU v3 devices that Google knew or should have known infringe one or more claims of the '616 patent. Google knew or should have known that there was and is a high probability that the TPU v2 and TPU v3 devices infringe the '616 patent. Alternatively, Google was willfully blind to such facts.

111. At the time that Google began using the TPU v4 device in the United States, Google knew or should have known that the accused TPUs incorporated one or more of Dr. Bates's patented inventions claimed in the '616 patent. Google knew or should have known that there was and is a high probability that the TPU v4 devices infringe the '616 patent. Alternatively, Google was willfully blind to such facts.

112. By the date of issuance of the '616 patent, due to its knowledge of Singular's Infringement Contentions served in case No. 1:19-cv-12551 (D. Mass.) involving patents related to the '616 patent, Google knew or should have known that the accused TPUs infringe one or more claims of the '616 patent and/or that there was a high probability that the accused TPUs infringe the '616 patent. Alternatively, Google was willfully blind to such facts.

113. In view of, *inter alia*, its: (1) involvement in the ongoing patent litigation with Singular in this Court, including its knowledge of Singular's Infringement Contentions therein; (2) IPR petitions regarding patent applications and patents owned by Singular; and (3) close monitoring of Singular's patent portfolio, Google knew or should have known since at the latest on or around August 25, 2020 that there was a high risk that the accused TPUs infringe one or more claims of the '616 patent. Alternatively, Google was willfully blind to the fact that such devices directly and/or indirectly infringe one or more claims of the '616 patent.

114. Google's actions described herein regarding the accused TPUs constitute conduct that was and continues to be willful, wanton, malicious, in bad-faith, deliberate, consciously wrong, flagrant and/or characteristic of a pirate. Google's egregious conduct has continued unabated since the issuance of the '616 patent.

115. Google's infringement of the '616 patent and willful conduct described above will continue unless and until Google is enjoined.

116. As a result of Google's infringement of the '616 patent, including its egregious and willful conduct described above, Singular has been irreparably harmed and suffered damages in an amount to be determined at trial.



**PRAYER FOR RELIEF**

WHEREFORE, Singular requests that the Court:

- A. enter judgment in favor of Singular on both counts of the amended complaint;
- B. award Singular damages resulting from Google's past and ongoing infringing conduct, together with interest thereon and costs pursuant to 35 U.S.C. § 284;
- C. award Singular enhanced damages pursuant to 35 U.S.C. § 284;
- D. award Singular its attorney's fees incurred herein pursuant to 35 U.S.C. § 285;
- E. enjoin Google's infringement of the '616 patent and the '775 patent pursuant to 35 U.S.C. § 283, and
- F. award Singular such other and further legal and equitable relief as the Court may deem just and proper.

**DEMAND FOR JURY TRIAL**

Singular demands a trial by jury on all issues so triable.

Dated: March 10, 2022

Respectfully submitted,

/s/ Paul J. Hayes

Paul J. Hayes (BBO #227000)

Matthew D. Vella (BBO #660171)

Kevin Gannon (BBO #640931)

Brian M. Seeve (BBO #670455)

Daniel McGonagle (BBO #690084)

**PRINCE LOBEL TYE LLP**

One International Place, Suite 3700

Boston, MA 02110

Tel: (617) 456-8000

Email: phayes@princelobel.com

Email: mvella@princelobel.com

Email: kgannon@princelobel.com

Email: bseeve@pricelobel.com

Email: dmcgonagle@princelobel.com

ATTORNEYS FOR THE PLAINTIFF

**CERTIFICATE OF SERVICE**

I hereby certify that all counsel of record who have consented to electronic service are being served with a copy of this document via the Court's CM/ECF system.

/s/ Paul J. Hayes